ANÁLISIS DE IMÁGENES EN CORREOS SPAM

Isaac Alonso Pérez
I.T.T.: Sistemas de Telecomunicación
Proyecto Final de Carrera

Tutor: Ángel Navia Vázquez
Dpto. Teoría de la Señal y Comunicaciones

ÍNDICE

1. Presentación del proyecto

En la actualidad una de las formas más comunes de comunicación entre personas son los correos electrónicos o emails. Pues bien, igual que este método de comunicarnos puede ser utilizado lícitamente, también puede serlo ilícitamente. El envío de SPAM en correos electrónicos es una de estas formas ilícitas.

Dentro del envío de SPAM en correos electrónicos existen diversas formas aunque en el proyecto nos centremos en el compuesto por imágenes.

Estas imágenes pueden ser clasificadas como SPAM siempre que el sistema para detectarlo funcione correctamente. Pues bien, estas imágenes suelen estar alteradas para que los sistemas no se han capaces de distinguir entre imagen con SPAM o sin él.

Entonces, en el proyecto presentaremos una clasificación de las alteraciones introducidas en las imágenes, así como el porcentaje de éxito alcanzado en nuestro sistema de detección de SPAM con los archivos originales.

Tras esta pequeña clasificación, pasaremos a realizar procesados sobre las imágenes de partida para aumentar el éxito de nuestro sistema. No sólo buscaremos este aumento sino que además llegaremos a fijar una serie de procesados modelo para cada tipo de modificación.

Por último, presentaremos las conclusiones alcanzadas con nuestro proyecto y futuros trabajos a realizar, teniendo como partida los resultados alcanzados con éste.

2. Introducción al SPAM

2.1 ¿Qué es el SPAM?

Según el Ministerio de Industria, Turismo y Comercio el SPAM se define como "*mensajes, habitualmente de tipo comercial, no solicitados y enviados en cantidades masivas. Aunque se puede hacer por distintas vías, la más utilizada entre el público en general es la basada en el correo electrónico*."

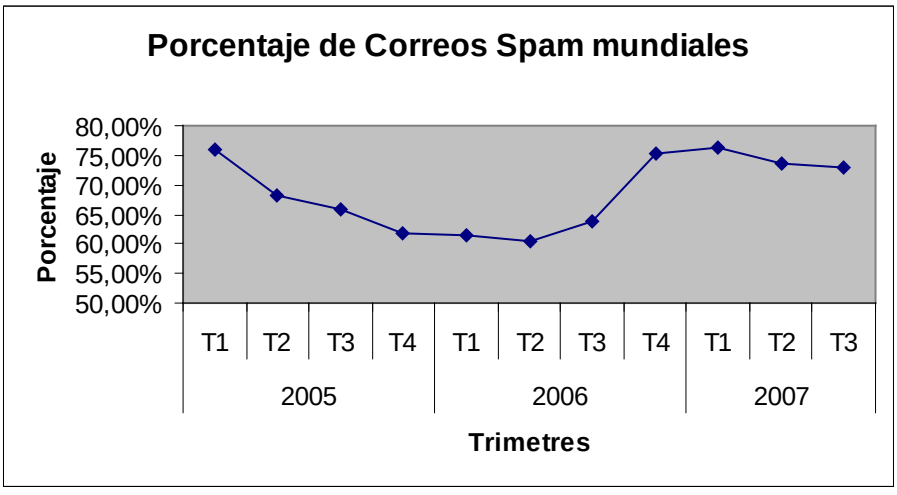
Por lo tanto, el SPAM se corresponde con el envío masivo de mensajes que son transmitidos al receptor sin haberlo solicitado, por lo tanto perjudican de una u otra manera al usuario.

Este tipo de piratería, prohibida mediante leyes estatales, se distribuye por distintos medios, que abarcan desde programas de mensajería instantánea hasta telefonía móvil, pasando por juegos en línea y correos electrónicos (Emails). La distribución de SPAM por correos electrónicos será nuestro punto de atención en este proyecto.

2.2 Cifras sobre el SPAM en correos electrónicos

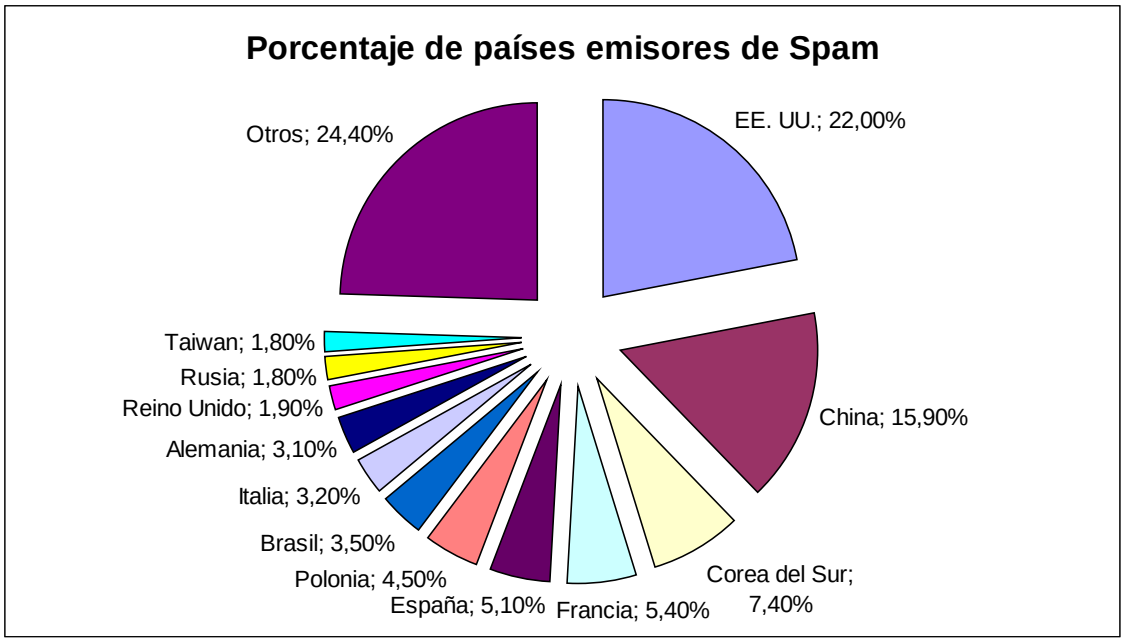
En este apartado se realizará un breve estudio sobre el SPAM en los correos electrónicos, teniendo en cuenta la procedencia de estos mensajes y el número de ellos que son generados habitualmente.

Por un lado cabe destacar la evolución de este tipo de mensajes frente a los mensajes enviados de forma lícita. En un estudio realizado hasta el tercer trimestre de 2007 se aprecia la siguiente evolución:

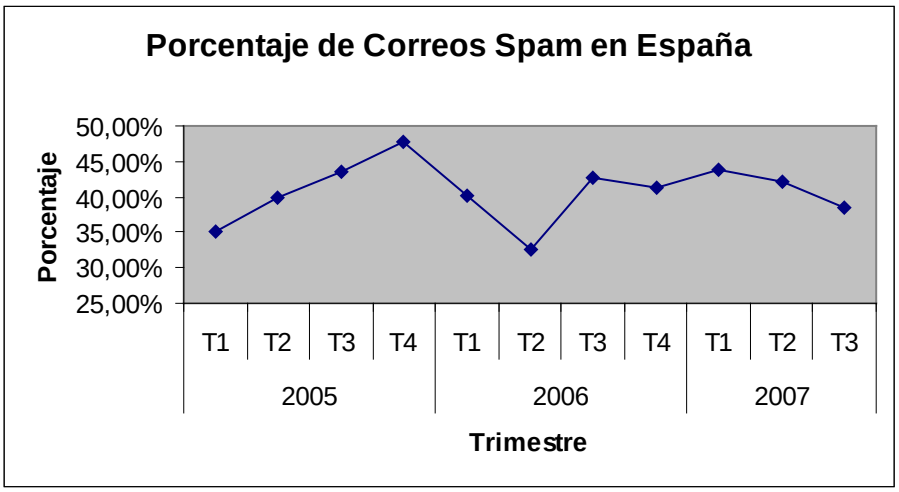


Se observa que, desde principio del año 2005 hasta mediados del 2006, el porcentaje de correos SPAM sufría un descenso. Sin embargo, en los dos últimos trimestres del 2006 aumentó el número de mensajes SPAM hasta comienzos del año 2007. A partir de ahí, se mantiene el porcentaje en torno al 75%.

Por otra parte si tenemos en cuenta los países con mayor porcentaje de emisión de correos SPAM, en primer lugar se encuentra EE.UU., seguido de China. Cabe destacar la quinta posición que ocupa España, siendo el segundo país europeo en esta clasificación.



De cara al porcentaje de correos SPAM en España, la mayor subida se produjo a lo largo del año 2005 mientras que el menor porcentaje se alcanzó en el segundo trimestre del siguiente año. En el año 2007 la proporción se estabilizó, cercana al 40 %.



2.3 *Evolución de las técnicas de SPAM*

Las técnicas de SPAM en los correos electrónicos se podrían clasificar por el lugar en él que se introduce éste, es decir, en el propio lenguaje http del mensaje (cabecera, título…) o bien en el propio contenido del correo. También existen listas

negras o blancas en las que se filtra el mensaje según en la lista en la que se encuentre el emisor de éste. Nosotros nos centraremos en el introducido a nivel de contenido, obviando el resto de los casos.

SPAM en modo texto

En los comienzos, todo el SPAM era enviado en formato texto. Este primer tipo consistía en un texto plano con publicidad, el cual era escrito directamente en el contenido del mensaje.

La siguiente evolución consistió en deformar el propio texto para complicar la identificación de ese correo como SPAM, intentando equivocar a los filtros de detección pero haciendo que para el usuario siga siendo igual de legible que anteriormente. Estas técnicas son muy diversas, a continuación se muestran algunas de las más importantes:

- Separar con espacios las palabras claves a codificar. Usar “S P A M” en vez de “SPAM”, por ejemplo.
- Insertar otro tipo de carácter que no sea el espacio: “S-P-A-M” o “S*P*A*M”
- Realizar cambios en dichas palabras: “SP@M” o “Publ1c1dad”

Se puede ver que estas técnicas son muy variadas y pueden cambiar muy rápidamente por lo que complican mucho la clasificación de estos correos como SPAM.

Con las dos técnicas anteriores, la clasificación entre correo lícito o ilícito se realiza mediante filtros. Al comienzo surgieron los filtros heurísticos que se basan en la propia experiencia del que sufre ese tipo de correo. Por ejemplo si se detecta que la mayoría de los emails no deseados tienen la palabra “Publicidad”, se genera una nueva regla en la que se propone que todos los mensajes que contengan esa palabra sean eliminados por el filtro.

El mayor problema de los filtros heurísticos antispam es que son estáticos. Las reglas para los filtros no son actualizadas automáticamente, por lo que resulta casi imposible tener una eficacia alta debido a la facilidad de saltarse el filtro con sólo modificar un poco el mensaje.

A continuación se optó por realizar una clase de filtro que fuera adaptativo, es decir, que se encargara de actualizarse automáticamente. Por eso surgieron los filtros Bayesianos, que se encargan de “aprender” de los mensajes que han sido clasificados anteriormente como SPAM y así aplicar estas reglas en los mensajes futuros. Para ello, hay una primera fase de aprendizaje del filtro con ejemplos que se clasificarán manualmente para fijar unas reglas con las que comenzar. Tras esta fase, a mayor uso del filtro, mayor eficiencia alcanza ya que aprende nuevas reglas que no hemos introducido manualmente.

En los filtros Bayesianos, la probabilidad de que una palabra sea considerada como SPAM viene dada por la siguiente expresión matemática:

$$\Pr ob = \frac{P_1}{P_1 + P_2}$$

siendo P_1 la probabilidad de que esa palabra aparezca en un correo SPAM y P_2 la probabilidad con que esa palabra aparece en correos legítimos. Por ejemplo, la palabra "OFERTA" está presente en 30 de 250 correos SPAM y en 25 de 100 correos lícitos, la probabilidad de que un correo que posea la palabra "OFERTA" sea ilícito será del 82 %.

$$\Pr ob = \frac{30/250}{30/250 + 25/1000} = 0.82$$

SPAM usando caracteres para dibujar el mensaje:

La siguiente técnica llevada a cabo en la creación de correos no deseados fue la de "dibujar" en el contenido del mensaje, las palabras que se quieren transmitir. En este caso el éxito depende mucho de la habilidad de los creadores.

La técnica de detección se basaría otra vez en filtros estadísticos aunque el uso del "dibujo" en SPAM es bajo, debido a lo laborioso que resulta y a que los mensajes a transmitir son cortos por razones obvias de espacio.

SPAM en imágenes:

El SPAM por imágenes sigue existiendo un mensaje de texto para el usuario. Sin embargo, la principal diferencia radica en que, en el SPAM con imágenes, como su propio nombre indica, éste es introducido en un archivo gráfico o imagen formada por palabras integradas en ella.

Estas imágenes son desplegadas finalmente por el usuario sin ningún problema, de modo que éste las encuentra totalmente legibles, mientras que los programas de detección de SPAM hasta ahora estudiados no son capaces de detectarlas.

En la siguiente imagen se muestra un ejemplo de SPAM:

Hasta la aparición de este tipo de SPAM, eran necesarios una serie de filtros, con mejores o peores prestaciones pero que trabajaban directamente con el texto a clasificar. Ahora, en este caso, ya no es posible analizar únicamente con filtros porque nos encontramos con imágenes y se hacen necesarios otro tipo de complementos.

Este complemento es un sistema de reconocimiento de caracteres (OCR). Éste consiste en que, de una imagen, es capaz de sacar el texto que contiene y convertirlo en una cadena de caracteres, la cual será la que se introduzca en los filtros para la detección.

Este procedimiento de transformación de imagen a texto comienza con el aislamiento de la parte de la imagen correspondiente a cada carácter y la compara con una base de datos de caracteres para identificar éste. Dependiendo de esta base de datos, el OCR tenía mejores prestaciones a la hora de distinguir distintas letras.

En la evolución del OCR, la comparación ya no se realiza sobre las matrices de cada carácter sino con las características básicas de los caracteres para facilitar el funcionamiento de éste.

A continuación se puede ver un ejemplo del funcionamiento del OCR, como a partir de un archivo gráfico obtiene un texto plano con la información contenida en la imagen:

=> PRUEBA DE UN RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR)

Sin embargo, la utilización de los sistemas de OCR posee una serie de inconvenientes. Para optimizar al máximo el funcionamiento de éste, la calidad de la imagen ha de ser lo mejor posible, sino es así, el sistema será incapaz de reconocer el texto. Otro de sus problemas es que el coste computacional es alto ya que tiene que comparar cada conjunto de puntos con las entradas en la base de datos de caracteres, lo que hace que el trabajo sea lento.

La siguiente evolución del SPAM en imágenes consistió en aprovecharse de las limitaciones existentes en el módulo OCR. Como la eficacia depende de la calidad de la propia imagen en la que se encuentra el texto a filtrar, basta con modificar la imagen de cualquier forma para que el OCR empiece a fallar. Seguidamente se enumeran algunas modificaciones en las imágenes con las que el OCR baja su efectividad:

- Introducción de ruido aleatorio a la imagen.
- Dividir la imagen en subimágenes.
- Modificación de los colores en el texto.

Se puede observar como las modificaciones con las que alterar la imagen son infinitas, por lo que para una correcta detección serán necesarios procesos en los que reconstruir la imagen original.

El paso posterior en la detección de SPAM en imágenes, ya no se limita al texto que contiene ésta, que a su vez era filtrado para conocer la legalidad del correo electrónico. En la actualidad para conocer si un archivo gráfico es SPAM, no se observa únicamente el contenido de la imagen, sino que se analiza las propiedades intrínsecas de la imagen.

El anterior método se puede resumir en comparar nuestra imagen a clasificar con una base de datos de imágenes que han sido clasificadas como SPAM. Dependiendo de la distancia o diferencia entre las imágenes, ésta será clasificada como SPAM o no. De esta forma ya no depende de la detección del texto introducido en el archivo sino de la propia imagen en sí y de sus características.

Gracias a la comparación entre imágenes se pueden evitar los problemas que presenta el OCR y se aumenta el ratio de detección de SPAM.

SPAM en archivos excel y pdf:

Una vez que la detección de SPAM en imágenes estaba siendo bastante satisfactoria, surgió el último de los tipos de SPAM. En este caso, consiste en introducir el SPAM en un archivo excel o bien insertar una imagen en un archivo pdf para evitar los filtros actuales de detección.

Aunque está siendo empleado en la actualidad el SPAM anterior, el modo que ocasiona más problemas sigue siendo el SPAM en imágenes, de ahí que nos ocupemos de forma extendida de este problema.

3. SPAM en imágenes

El apartado previo trataba de dar al lector una pequeña perspectiva a cerca del correo SPAM y su evolución a lo largo del tiempo. En la actualidad, la mayor parte del SPAM que reciben los usuarios en su correo electrónico es de tipo visual, es decir, incluye una imagen en la que se introduce la información que percibe el usuario.

Como se comentó anteriormente, en la primera fase del SPAM a través de imágenes, éste consistía en añadir un texto plano en una imagen. Por lo tanto el filtro para conocer si se trataba de un mensaje legal o ilegal se realizaba con un simple OCR que se encargaba de transformar la imagen a texto y que este texto fuera la entrada del decisor bayessiano.

Debido a que con el Reconocimiento Óptico de Carácteres se llegaba a un gran porcentaje de acierto de cara a la detección de SPAM, las imágenes que contenía este tipo de correos se fueron modificando y complicando para intentar que el OCR fallara en su cometido.

3.1 Transformaciones sufridas por las imágenes

En este nuevo apartado del proyecto, se pretende abordar los distintos tipos de modificaciones o transformaciones que han ido sufriendo las imágenes de cara a encontrar el error en la detección de SPAM.

Las modificaciones que se han ido realizando sobre estos archivos gráficos son de muy diversa naturaleza. Se basan en complicar la imagen, es decir, cambiar todo lo posible el archivo original de forma que el mensaje sea legible por parte del ser humano pero imposible por parte de las máquinas (OCR).

En cuanto a las técnicas de estas imágenes para generar SPAM, se pueden dividir en categorías. En la primera de ellas se trata de generar modificaciones sobre el plano principal de la imagen. La segunda consiste en aplicar tantas modificaciones como se quiera en el plano secundario o fondo y la tercera, en el formato. Así con cada modificación, una nueva imagen SPAM es creada.

Gracias a la búsqueda realizada en los correos no deseados, así como a través de internet, se han localizado gran cantidad de imágenes clasificadas como SPAM. Tras un estudio previo, se pasará a clasificar distintos métodos y modificaciones que se realizan con cada una de las tres técnicas.

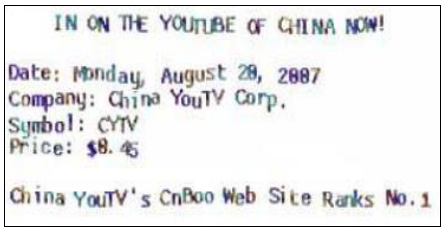
Transformaciones en el plano principal:

Previo a ninguna modificación, nos encontramos ante una imagen con un texto plano. Si hacemos cualquier tipo de cambio sobre la apariencia visual de este texto, siempre contenido en una imagen, estamos ante las técnicas de transformaciones en el plano principal.

Estas modificaciones pueden ser de muy distinto tipo, entre las que destacamos:

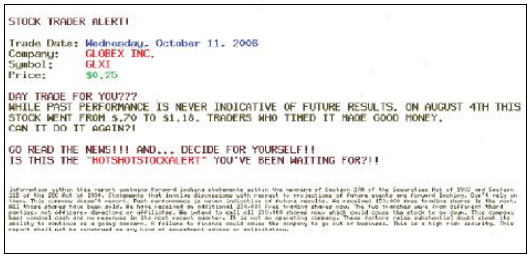
- **Rotación:** Equivale a realizar un giro sobre la imagen de partida. Se puede observar como se ha producido una rotación en el texto incluido.

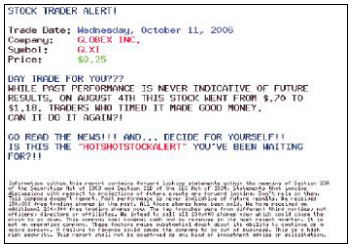
- **Ondas:** El texto sufre unas ondulaciones y consigue la forma de una sinusoide.



- **Texto deformado:** En este tipo de cambio se consigue que el formato del texto original no se parezca en nada al actual. Se modifican tanto el color de la fuente, el tamaño, el tipo de letra utilizada…

- **Estructura:** El texto incluido posee la misma información en distintas imágenes pero se puede observar como su estructura varía, cambiando tabuladores, espacios, sangrados…





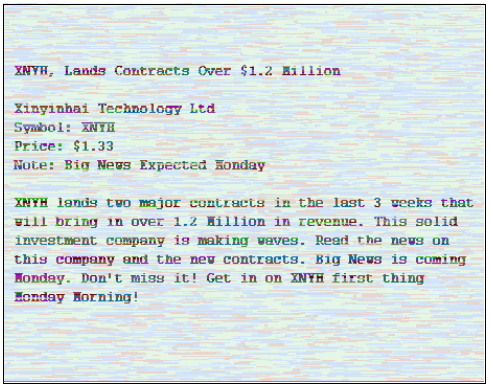
Transformaciones en el plano secundario:

Nos encontramos en el caso en el que únicamente se va a modificar el plano secundario o fondo de las imágenes.

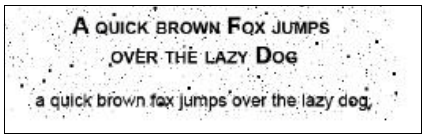
Hasta ahora todo cambio se había realizado sobre el plano principal, es decir, en el plano en el que se encuentra el mensaje que se quiere dar a conocer. A partir de este momento, los cambios se generarán en el fondo de la imagen por lo que no cambiará el texto pero sí la apariencia de donde está contenido.

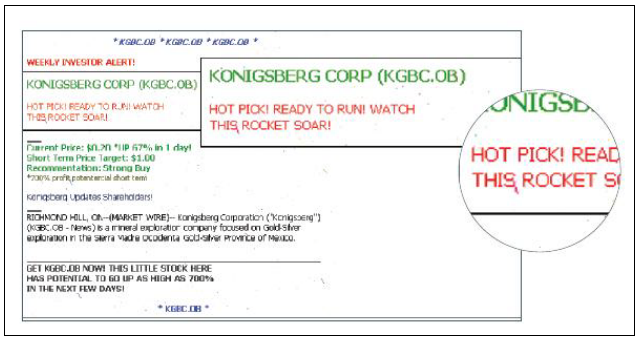
En este paso se han detectado gran cantidad de métodos para modificar las imágenes. Podemos centrarnos en los que se nombran seguidamente:

- **Colores:** Se intenta modificar los colores del fondo y usar varios de ellos para hacer nulo el trabajo de los OCR. Además, en la segunda figura, apreciamos que no se ha retocado todo el fondo sino sólo aquella parte que contiene la información.

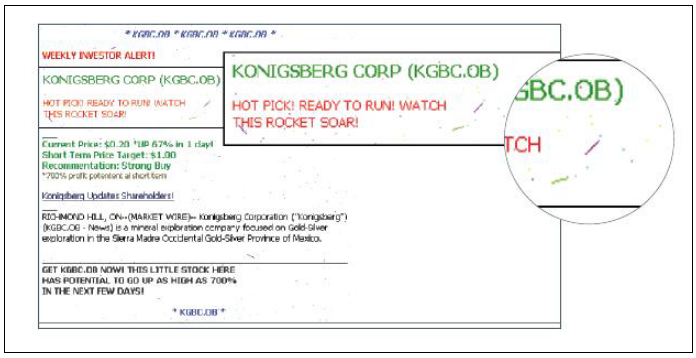


- **Puntos:** Esta técnica consiste en insertar puntos aleatorios en el plano secundario. Dichos puntos pueden ser, tanto de distinto tamaño como de distintos colores. De igual forma que ocurría en la imagen 2, modificando el color del fondo sólo donde se localiza la información, también se puede concentrar una mayor cantidad de puntos en esta misma zona de la imagen (*imagen 1*).

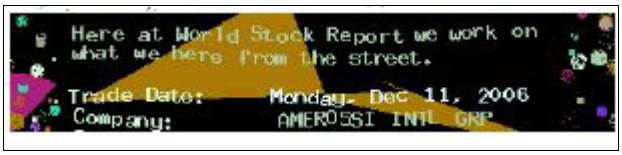




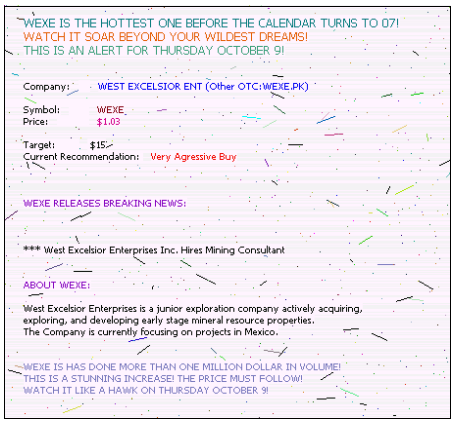
- **Líneas:** actúa de la misma forma que los puntos pero en este caso se insertan líneas en el fondo de las imágenes. Igual que ocurría previamente, se modifican colores, grosores, longitudes…



- **Formas aleatorias:** Se basa en el mismo principio que los dos tipos anteriores aunque las figuras a incluir en el plano secundario son más complejas. Los ejemplos siguientes muestran algunas de estas formas aleatorias.



- **Combinaciones:** Por otra parte, como era de esperar, también se pueden encontrar combinaciones de cada uno de los distintos métodos de modificar las imágenes. En la primera de ellas, se puede ver la inclusión de puntos y líneas aleatorios. En la segunda de las imágenes, se aprecia un fondo totalmente irregular en cuanto a colores y formas.



En resumen, la mayoría de los archivos con los que vamos a trabajar van a tener varios tipos de transformaciones y en pocos de ellos nos vamos a encontrar sólo un tipo de modificación en el fondo.

Transformaciones en el formato:

En todo lo comentado hasta este momento, únicamente se ha tenido en cuenta alteraciones en los dos planos existentes en las imágenes. Sin embargo, también pueden existir otros tipos de cambios referentes al formato exterior de la imagen. A continuación se muestra una serie de ejemplos:

- **Deformación:** consiste en expandir, comprimir… la imagen de forma que se parezca lo menos posible a cualquier otra imagen.

- **Corte:** utilizado para partir el texto que aparece en la imagen en dos o todas las partes que se quiera. El corte puede producirse tanto verticalmente como horizontalmente.

Transformaciones en el contenido:

Fuera de las transformaciones en los dos planos de las imágenes y en el formato de éstas, también hemos encontrado archivos en los que el contenido de la imagen ha sufrido cambios para que al ojo humano la información parezca legible pero para las máquinas no lo sea.

En la figura próxima mostramos cómo se juega con el color de la fuente del texto y el fondo de la imagen. De esta manera, quedan camufladas partes de las imágenes que no van a ser útiles para el usuario pero sí para intentar engañar al filtro anti-spam.

Transformaciones con varias imágenes:

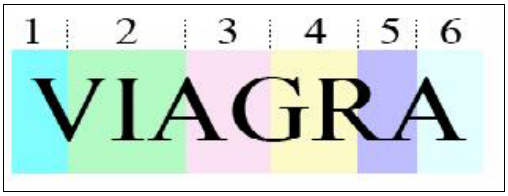
En las páginas previas se ha tratado sobre el SPAM formado por una sola imagen. En nuestra recopilación de imágenes, nos hemos topado con un nuevo tipo de SPAM gráfico: SPAM en varias imágenes.

Este un tipo de SPAM se basa en la utilización de varias imagen y no una sola, como era hasta ahora.

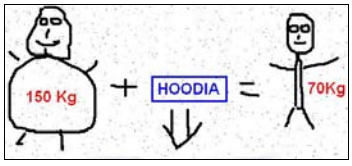
El funcionamiento es sencillo y sólo requiere subdividir la imagen principal en otros secundarias, de forma que el análisis anti-spam se hace con cada una de ellas independientemente y no se localiza ningún problema. Pero una vez reconstruida la imagen aparece de nuevo el SPAM.

El ejemplo grafico posterior pretende aclarar esta idea. La primera imagen es tras reconstruirla y las otras son las subdivisiones.

Otro posible ejemplo puede ser el siguiente, en el que la imagen original ha sido dividida en 6 imágenes de menor tamaño.



En conclusión con el apartado en el que nos encontramos, destacar que las modificaciones o cambios en las imágenes encontradas pueden ser muy diversas y que varían continuamente con una gran velocidad. Para reafirmar este hecho, se muestra la imagen siguiente, capturada de un correo electrónico.



De cara a tener una visión global del problema ante el que nos encontramos, en la primera etapa se hizo una pequeña introducción sobre el SPAM y se aclaró que nos centraríamos únicamente en el contenido en Imágenes. Una vez situados dentro de este tipo de SPAM, nuestros estudios posteriores, así como desarrollos, se localizarán en el SPAM gráfico de una sola imagen, sin prestar atención a aquel formado por un conjunto de éstas.

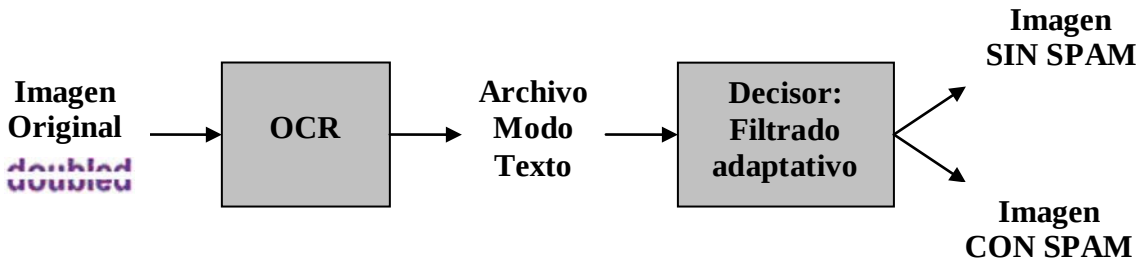
4. Análisis experimental: situación inicial

Hasta llegar al punto en el que nos encontramos del proyecto, nos hemos centrado en conocer características más generales acerca del correo no deseado que circula por la red y deducir cómo este correo puede contener imágenes. Además, en el apartado anterior también se vieron distintos tipos de modificaciones y cambios que sufren las imágenes para que puedan llegar a funcionar como SPAM gráfico.

En este apartado realizaremos en un análisis experimental sobre la situación de partida que tenemos con las distintas imágenes estudiadas hasta el momento.

Esquema de bloques del sistema de detección de SPAM gráfico con OCR:

El esquema de partida de cualquier sistema para la detección de SPAM en imágenes será similar al de la siguiente figura:



En este esquema previo se pueden visualizar los distintos pasos que se llevan a cabo para el análisis completo de la imagen:

1. Imagen original introducida al OCR sin ningún tipo de procesado previo.

2. Transformación del archivo gráfico a un archivo de texto gracias al OCR. Para ello el OCR transformará todos aquellos caracteres que reconozca en la imagen a un archivo en modo texto.

3. Decisor basado en filtros adaptativos cuya misión final es diferenciar entre imágenes que posean SPAM y aquellas que no lo posean.

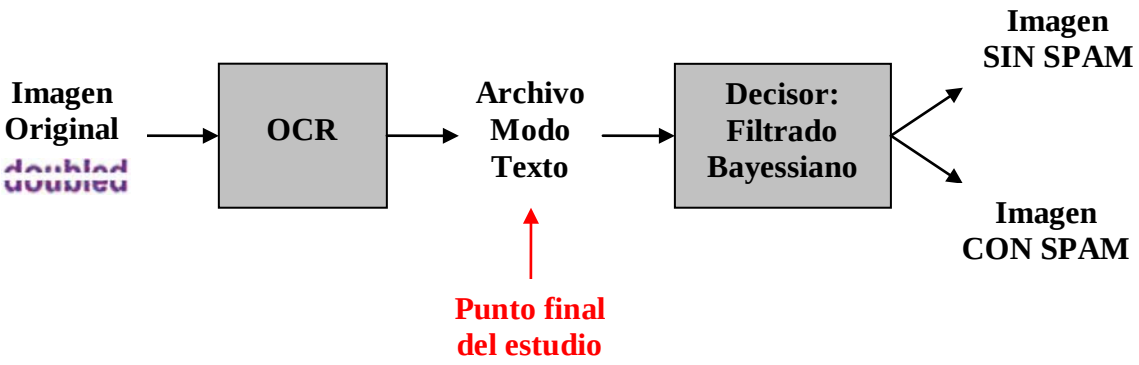
Experimento base:

Una vez que hemos detallado los bloques de los que se compone un sistema básico de detección de SPAM en imágenes, pasamos a simular experimentalmente un sistema como el anterior.

Para esta simulación se utilizarán varias imágenes de cada uno de los tipos nombrados en los apartados previos. El funcionamiento de la simulación será sencillo:

introducimos la imagen a estudiar en el OCR y éste nos facilitará un archivo modo texto.

Como se ha podido apreciar en el diagrama de bloques, tras el procesado llevado a cabo por el OCR, el archivo modo texto sería analizado por el sistema del decisor. Cualquier sistema de detección de SPAM acabaría con este último filtrado adaptativo, sin embargo, en nuestro estudio llegaremos hasta la obtención del archivo de texto. En el diagrama de bloques siguiente se marca el lugar en el que se detendrá nuestro estudio



En los sistemas de filtrado de imágenes SPAM como el del bloque anterior, la calidad se mide con la probabilidad de éxito del decisor, es decir, la probabilidad de que se decida correctamente si la imagen contiene SPAM o no. Sin embargo, en nuestro estudio no existe este decisor, por lo que la probabilidad que se menciona en ellos no tiene ningún sentido.

Para conocer el éxito o fracaso de nuestro sistema nos basaremos en la tasa de acierto del OCR, es decir, el porcentaje de letras recuperadas correctamente en el archivo de texto (generado por el OCR) respecto a las contenidas en la imagen:

$$Tasa\_acierto\_OCR = \frac{n^o\_carateres\_correctamente\_recuperados\_OCR}{n^o\_carateres\_totales\_imagen}$$

4.1 Estudio experimental: Resultados iniciales

En el punto anterior quedó definido en qué lugar del diagrama de bloques y respecto a que parámetro se medirá la calidad de nuestro sistema.

A continuación pasaremos a obtener unos resultados prácticos con imágenes que hayan sufrido los distintos tipos de modificaciones que se nombraron en el apartado de Transformaciones. Para ello, nos serviremos de cada imagen y por supuesto de un software que simule un OCR. El software utilizado para nuestra simulación será "*Softi FreeOCR*" aunque se podrá realizar con cualquier otro software que emule el funcionamiento de un OCR.

Los resultados prácticos se obtendrán para cada tipo de transformaciones que pueden sufrir las imágenes y con todos los ejemplos posibles que tengamos de cada una de ellas. De cara a que los resultados sean a la vez claros y fiables, se mostrará el

archivo gráfico original junto con el archivo de texto generado por el OCR. A partir de estos dos archivos se podrá generar la tasa de acierto del OCR que viene a ser nuestra medida de calidad del sistema de filtrado de SPAM.

Transformaciones en el plano principal:

En este subapartado trataremos de estudiar el funcionamiento del OCR sobre archivos que hayan sufrido cambios en el plano principal de ellos. Haciendo un pequeño recordatorio, los cambios más importantes son: rotación, ondas, texto deformado y estructura.

- **Rotación:**

Imagen original "Rotación1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
	Buy!	4	4
	BUY1	3	4
	BU\Buy!	6	7
nº Caracteres recuperados correctamente:		**13**	
nº Caracteres totales:		**15**	
Tasa de acierto del OCR:		**86,67%**	

Imagen original "Rotación2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
	CANADIAN MNERALS ARE AN UNTAPPED MRRKETE	33	36
	{WPS IS HERE T0 DIG OUT THAT GOLD!	24	27
	C0rnpany:IRWi1•I RESYXYRCEB INC (01hBr OTC:1wR 5 . P K)	28	43
	gxumhi W R 5	2	11
	Trading at: 0.9% {UP 11.11%}	24	25
	5-Day Est: $3.5G	13	14
	Ly Target Est; $15	13	15
	Mélkct indllatutz Bnlsh	11	23
	GET ON THIS BANDWAGON NOW!	22	22
	OLP- LAST FEATIRE GAINED 2{X}% IN A WK!	25	30
nº Caracteres recuperados correctamente:		**195**	
nº Caracteres totales:		**246**	
Tasa de acierto del OCR:		**79,27%**	

Imagen original “Rotación3”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	PREIWIER ‘F‘·H.¤`R'Ms3.CY` •· Lc·was't \!§P..f3F2.A, CIAL'13, LENHTRD. Onllns F·r"|ce! \!T.M3R.A BD '$Z—l34.Q'5 CIALIS CVD $L6'i}'95 VALKUM EU $B'5.4E» 'SOD-'lr1\ 30 *$'?5.9'5 |LF'RiUT"E.C.Es'+ SG *564.95 M-'nE»KEl\L EF] $12DSQ • ><aM.·¤>< 3D $123.45 ·a1.¤@·Rs>. (BOFY 5m $25D.99 L \New CKBEQSOW ED 522-·'r.2'§ rj xx Save up to $$0% mn your prescrimion Meds!	11	15
		23	41
		18	30
		19	26
		11	31
		18	34
		9	22
		30	34
		0	7
nº Caracteres recuperados correctamente:		**139**	
nº Caracteres totales:		**240**	
Tasa de acierto del OCR:		**57,92%**	

Imagen original “Rotación4”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Canadian Pharmacy 1'arycuf .r"€¥=}iZT`§§3T§""" - J; crm - ss:. P5 I! Wégra Sw? rains - $2.40 x _ _ _ / ffraws sca1'?¢'5*b,g — $5. F8 if __g_E¤ more ____`___`__w, The bw czuarfsy anu the éeg; cme; nay'! Wl ern in your banker; www. 4pharm. Nat	19	23
		2	12
		3	12
		11	20
		7	20
		4	7
		12	30
		9	29
		13	14
nº Caracteres recuperados correctamente:		**80**	
nº Caracteres totales:		**167**	
Tasa de acierto del OCR:		**47,90%**	

En este primer tipo de transformaciones, podemos apreciar como la tasa de acierto del OCR en la primera imagen se encuentra en un valor alto, cercano al 80 % aunque en las dos últimas imágenes baja hasta el 50%. Por otro lado, también se puede ver como la mayor tasa de acierto se produce cuando las letras de la imagen son de mayor tamaño.

- **Ondas:**

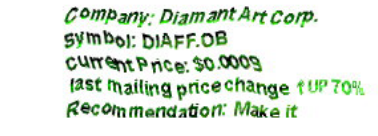
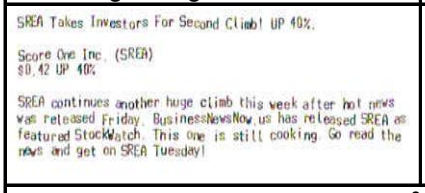
Imagen original “Ondas1”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	&‘°·*·'¤'»¤arr;n· Dimmfin carp- 5¥ 'mhusz DIAF F.0B cummv me: wa Nw lm m¤1un¤ page uname t up 10% RGCOH1 mandztfcrlf Make it	5	25
		7	15
		4	20
		8	28
		9	21
nº Caracteres recuperados correctamente:		**33**	
nº Caracteres totales:		**109**	
Tasa de acierto del OCR:		**30,28%**	

Imagen original "Ondas2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	ll cu THE Y0un_B£ cr cmm nm! Date: Ihnday, gqgust B, gg? cnqaang: China YnuTV €¤r'p. E-'"°$;°!:•¤CTg China Yami'; cnhcm Heh Siu Rats un. 1		
		8	25
		14	25
		15	23
		2	11
		0	11
		13	33
nº Caracteres recuperados correctamente:		**52**	
nº Caracteres totales:		**128**	
Tasa de acierto del OCR:		**40,63%**	

Imagen original "Ondas3"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	SER Takas {mentors Fur Elmnd Ciirbf UP 18%. Sgure Om: lm: [SRE) SDA? LIP IDR SRE9 mntinws mqihnr hugs climb this mask after hn! news up rgiguacd Friday B|,|lifIGS$k\'¥l1,I.I$ has_|·¤L¤as·¤d' TER av fusurm Stncldlamh. 111is ¤n¤ es still molcrng Gu Nd lhs naw and sat nn SEER Tuendnvl		
		21	40
		9	17
		31	10
		18	49
		24	53
		13	52
		13	24
nº Caracteres recuperados correctamente:		**129**	
nº Caracteres totales:		**245**	
Tasa de acierto del OCR:		**52,65%**	

Con este tipo de modificación, el éxito del OCR se aproxima en algunas imágenes al 40% aunque en el penúltimo caso, el porcentaje aumenta hasta un valor próximo al 52%.

- **Texto deformado:**

Imagen original "TextoDeformado1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Bullslzye Finarrcial Weekly Report Ser Make no mktake, our mission at BulhEye Finm mrderperformirg companies our there to find th The micro-eq: hnnnd that can make you a for profile show asignificant increase in stock price or ars. Wgchave come across what we feel is me of thas about yet. Trade Date: Tuesday, September 5, 2006 Company : TRIMAX CORPORATION Ticker : TMXO Current Price : $0.38 Short Term Target Price : $1.50 Long Term Target Price : $2.50 Recommendation: STRONG BUY		
		29	31
		33	38
		36	40
		29	37
		43	43
		6	8
		33	36
		9	9
		33	33
		25	25
		11	11
		18	18
		26	26
		25	25
		24	24
nº Caracteres recuperados correctamente:		**380**	
nº Caracteres totales:		**404**	
Tasa de acierto del OCR:		**94,06%**	

Imagen original "TextoDeformado2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	@·EI} DRUGS LOWEST ONLINE PRLCE GU.¤•RAN'|`EED! UIAGRA CIALIS LE\II'I'RA $138 $3.00 $3.33 Ne guamntee1£I]% TOP-QUAUTY ef the product we effec! GET 4 VIAGRA QLLS FREWTH ANY ORDER!		
		7	8
		25	28
		15	19
		10	12
		36	46
		28	32
		0	34
nº Caracteres recuperados correctamente:		**121**	
nº Caracteres totales:		**179**	
Tasa de acierto del OCR:		**67,60%**	

En este caso de transformaciones, la tasa de acierto del OCR depende, en gran medida, de la cantidad de modificaciones que haya sufrido la imagen. Esta tasa varía, desde el 68% del segundo archivo, hasta un acierto casi completo con el primero.

- **Estructura:**

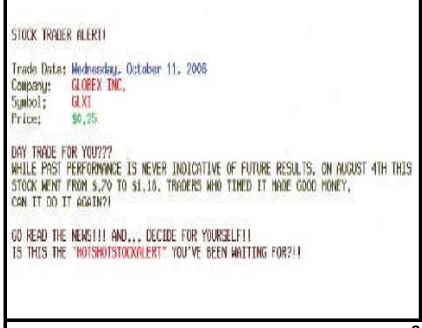
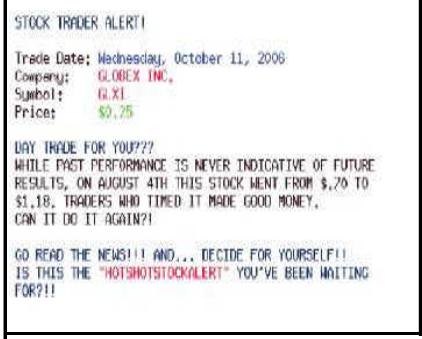
Imagen original "Estructura1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	SIUEX IRFI.I£I< IltkII Irade [bits: IIs·<t¤¤¤?¢|¤u- Otzbnhcr 11. ?C•¤·B Culgq: HEHE! DI;. k `I: |3I.>(I Prizm: U.?5 DR'! TRFDE FDR `|'DU'??':' IIIILE PH5¥ FERFORIQWICE IS IE'¤*ER DDIGQTIVE OF F¥.I'IUFE REEL!. FS. ON H.ICI.\$T ·ITH THIS STOCK ENT FIIIH $.70 'ITI 51.10. THERE HPI} 1'IIED 1T FII! DOG} IDM?. CMI IT DU IT MATNQI GD IEFI] IIE DEIISIII HD". DECIDE FUR WJLIIELFII [5 THIS ITE 'I|DT$'KJT31CK}<FI.ER!' TEIJ'\F'E BEEN IGITIIG F|JR?*!		
		2	17
		11	25
		4	18
		0	11
		6	11
		10	17
		38	66
		26	54
		6	16
		16	41
		20	47
nº Caracteres recuperados correctamente:		**139**	
nº Caracteres totales:		**323**	
Tasa de acierto del OCR:		**43,03%**	

Imagen original "Estructura2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	$TGZ|< TRHIH HLERTI Irene Date: rlamasclag. Gnuhnr 1J., 20015. C-mvuvg: UJIILI {NZ:- °»u»Iml: u.J<1 Prlcet W. PF- LIRY {INI}. FOR YEILIZW? IIIILE PREV FERFOFUIANGE IG IIVIIR TNJICATIVE 0I' FIJTLRC ESLIJS. ON NJGUST 4TH THIS S`|'|ZI21< Q-!-LN7 FRCII S.?0 I0 ¢I.I8. `I'Rf-'IZERS iIIII TIEU IT HMI BDU HIIIEXH UN I'! I1} IT QGHIII?1 GD REQD `ITE IE|.¢'3!'! IJEUIDE FCE YDLRELFN I[gR;|H5 TPE `HI¥I`5I||J!\'EICK3|&fI.ERI` YU.|'VE BEEN WIIIING		
		6	17
		14	35
		3	11
		0	11
		4	11
		6	17
		24	45
		21	42
		11	37
		6	16
		16	42
		18	45
		0	6
nº Caracteres recuperados correctamente:		**129**	
nº Caracteres totales:		**335**	
Tasa de acierto del OCR:		**38,51%**	

Únicamente realizando cambios en la estructura de la imagen se puede apreciar que los resultado en ambos archivos son muy próximos Este hecho era de esperar ya que partimos prácticamente del mismo texto en la imagen, pero en el que se han modificado tabuladores, espaciado… modificando muy poco la tasa de acierto del OCR.

Transformaciones en el plano secundario:

En el apartado previo habíamos trabajado sólo con modificaciones en el plano principal de las imágenes. A partir de este momento, las transformaciones se realizarán sobre el plano secundario, es decir, variando las propiedades del fondo de las imágenes y observaremos el acierto que es capaz de alcanzar el OCR con estos cambios.

- **Colores:**

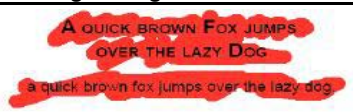
Imagen original "Colores1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
	R nédwm F0: JLIJF5 -	2	17
	`{`cvvn ru; |.AzYDO¢; _	5	14
	a tpdé bun-».n P¤x_|un1ps uvqr 1m lazy dag.	14	34
nº Caracteres recuperados correctamente:		**21**	
nº Caracteres totales:		**65**	
Tasa de acierto del OCR:		**32,31%**	

Imagen original "Colores2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
	Buy	3	9
	ViAgRa	6	6
	S 3.99 V'AgR"	8	11
	ViAgRa	6	6
nº Caracteres recuperados correctamente:		**23**	
nº Caracteres totales:		**32**	
Tasa de acierto del OCR:		**71,88%**	

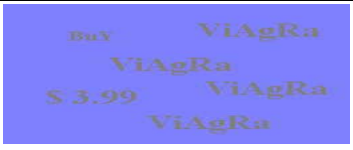
Imagen original "Colores3"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
		0	9
		0	6
		0	11
		0	6
nº Caracteres recuperados correctamente:		**0**	
nº Caracteres totales:		**32**	
Tasa de acierto del OCR:		**0,00%**	

Imagen original “Colores4”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	=., ;.-.-··¤=¢..€—- ai¤r.:·.:;::: mr: ;,=..; r··1.22··e;;		
	i1a:·1=.=a1 ‘‘:’s;c!·.·w=.:E-.—·····r Lrn	4	34
	:gni.u.‘|E:	1	22
	51. 4*	1	11
	i..;.;;: 51; :.1;:.: EE;::,..‘.;4.&x; ;;:*;r¤·;r	2	11
	EL3; Lim-; ‘Tr L-·;?··· =·=...i..... --; ’;”¤ ’¤E” `Z	1	26
	TZEE Zi?-T	2	45
	211; ==·‘1:1.1 ‘ -- - ;’¥i¥` --.. 2.. Laxman: -`T-MZ ...4-;	2	44
	:.:._.:.:::__:n_.!!!_ unnuz;-··¤_¤ 1q unl’! FF- -: 1}... ..	1	44
	-I 1..	1	45
	iiiz :::, ..,. ...3; ==1 ini EEE .::2;-- - - ":.: 2--- -- ::1;;**	1	39
	;-.....v. 2..:’I ¤¤€=‘- -2; ain; Lu uu ;Z...: $1:2*.: Zihinu	2	14
nº Caracteres recuperados correctamente:		**18**	
nº Caracteres totales:		**335**	
Tasa de acierto del OCR:		**5,37%**	

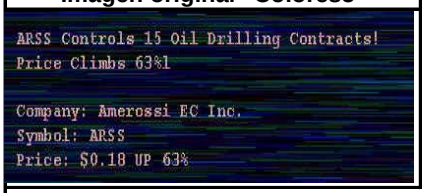
Imagen original “Colores5”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
	ARSS Ccmtr0ls—15 Oi-L-’briiiing Contracts.!	28	35
	_ PricE"C].imhs 639as]." ’ ` `	10	15
	`Ccrmpa·ny·:` Amcrussi .E.C. II1!Z€.» _	15	22
	"S$éiii1i¤.1`= mss	3	11
	Price: $0.13 UP 63%	15	16
nº Caracteres recuperados correctamente:		**71**	
nº Caracteres totales:		**99**	
Tasa de acierto del OCR:		**71,72%**	

Imagen original “Colores6”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
		0	17
nº Caracteres recuperados correctamente:		**0**	
nº Caracteres totales:		**17**	
Tasa de acierto del OCR:		**0,00%**	

Como se ha podido apreciar, modificando el fondo de los archivos a estudiar, obtenemos porcentajes de éxito del OCR de muy diversos valores. Estos valores dependen en gran medida de la diferencia de contraste entre el plano secundario y primario. A mayor grado de similitud entre los dos planos (imagen *Colores3*), menor tasa de acierto. Por lo tanto, la tasa de acierto es inversamente proporcional a la diferencia de contraste entre planos.

- **Puntos:**

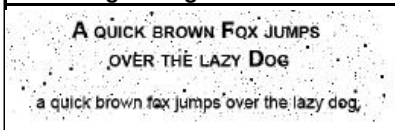
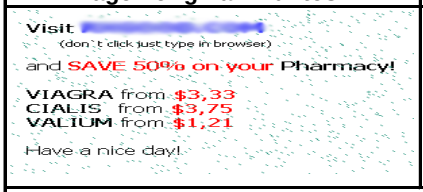
Imagen original "Puntos1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	. .".- Qéncklwandxkjn Fdfldmmé Y. ‘ i .pv;R-nié·u5zv_D¤q ·-·_ ’. Y `fa é|`ul;_ré ¤F¤wn_f5x‘]QrqpE'6vé?_thé_j'l_a$y E1»j;._. ¤		
		2	19
		2	14
		6	33
nº Caracteres recuperados correctamente:		10	
nº Caracteres totales:		66	
Tasa de acierto del OCR:		15,15%	

Imagen original "Puntos2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	\\:_ {d0nE§`cHpI$·{ug[;tl·bg i:H·brgw%e¤]! 0: ,2. \ ~` §¤q_sAk{n; $0% `;s¤rv0u{`m¤a&m$6g4j * Q`v1AGhA`%&5m.—‘$3Q3§`_ — Q _ `CI2'\|;1$_ ¤°KQm;`$3.75 ~. ~ Wk[;TUM—F(0m.2$1;2} {}:_;:;;fL I ‘ ` I~E|ax·5§‘@`miEé d@Q!.;‘ —	0	5
		10	29
		5	25
		7	13
		8	13
		6	13
		4	13
nº Caracteres recuperados correctamente:		40	
nº Caracteres totales:		111	
Tasa de acierto del OCR:		36,04%	

Introduciendo puntos dentro del fondo de los archivos que estamos estudiando, se puede lograr que el rendimiento del OCR descienda en ambas imágenes.

- **Líneas:**

Imagen original "Líneas1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	at >—q ~>»<`?T»<» M/;»~ ;.— N jf »€ v my {WIA·-QljlCK*BR0WN{F0X|JUMP§ I \ '<\x\,~<—*\ .,\'¤'I, —~.\ §l,·'\\/_____hi-.Q;/`\\\J,·**·\Q/)'}</\\'¤ _ ._ _ (\ { / * , ><` A -" X gs}/';_3y} |0jfER\THE,LAZ;DOG ij \j/ \` Ek/\` ~ x/\ */!73 > \"»7`.(g/ '¤` .·=—<·'fI ')3`*quiclg(br0wr§.foxffjum?s—-.0ver,the,laiy'd09\,»\ `>\ /`*\\\/~—-\\ \£" \\—fz>/		
		14	19
		10	14
		23	33
nº Caracteres recuperados correctamente:		47	
nº Caracteres totales:		66	
Tasa de acierto del OCR:		71,21%	

Imagen original "Líneas2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	*;TOri?.A` * *`T01{A~*€ T0i1A` *` " ~ ' \ ~¤—¤ `W/IM »*, \ I X \ / N /· _ \ / · » \ CompanyfName: / TORAYFECHNOLOGIES |(0TC '?BB:TORAi)B) `?2EE"§;°"-?i\;13T%%` —,\— j xx 0 ¤ ff éfday/Target: $4"· ¤|_. II \ Y x \ \Gf1rr'ent j_Vlarket: '\\»'eryEuIlish \ ·r/ I f ' / ff Recommendation: \ Strong Buy I { \ " / / ¤\»/\\/\\/_"V/\\		
		8	12
		39	43
		1	11
		0	14
		12	14
		20	25
		24	24
nº Caracteres recuperados correctamente:		104	
nº Caracteres totales:		143	
Tasa de acierto del OCR:		72,73%	

Observando los resultados alcanzados tras la inserción de líneas en el segundo plano de la imagen podemos ver que la tasa de acierto del OCR está cercana al 70%. Podríamos pensar que es un buen porcentaje, pero es una cifra engañosa ya que si vemos los caracteres que devuelve el OCR, existe un número mayor que los que contiene la imagen en si. Muchos de los caracteres que genera el OCR son generados debido a la confusión entre líneas y caracteres.

- **Formas aleatorias:**

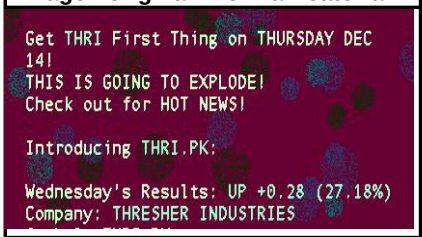
Imagen original "FormaAleatoria1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Get THRI First Thing on THURSDAY DEG 14l THIS IS GOING T0 EXPLODEI Check out for HOT NEWS! Introducing THRI.PK: Wednesday's Results: UP +0.28 (27.18%) G0mpany:_THRESHER INDUSTRIES		
		29	30
		3	3
		21	21
		19	19
		19	19
		34	34
		24	26
nº Caracteres recuperados correctamente:		**149**	
nº Caracteres totales:		**152**	
Tasa de acierto del OCR:		**98,03%**	

Imagen original "FormaAleatoria2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	5 ui-L-ru l".- HW: I- `k Renxwr. wr wvrk UH " ,5,.* ' '· . - ze n: ;:|.m¤1.. Q-,· vg E'*'°'° "*"*=¤ www- mc 11. zoos , · ""F'3¤ : DM!-k0'.t-il LN;. .;.·¢· ".		
		8	30
		1	24
		4	27
		4	23
nº Caracteres recuperados correctamente:		**17**	
nº Caracteres totales:		**104**	
Tasa de acierto del OCR:		**16,35%**	

En el segundo plano de las imágenes los cambios que se pueden realizar son de formas muy diversas. Con los dos archivos previos, las tasas de aciertos del OCR difieren de manera muy significativa. Esta diferencia se debe a las diferentes formas que han sido añadidas en el segundo plano de cada una de ellas, que hacen que los errores en el OCR aumenten según el tipo.

- **Combinaciones:**

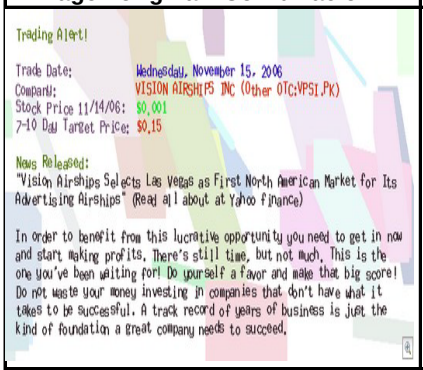
Imagen original “Combinación1”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Tn-adina me-1.! Trada Date: Hedne8das.|, November 15- 2006 comoam: VISION BIRSHIPS llc {Other 0Tc;vP51,PK) 5T·¤cls Price 11¢'14/06: $0-001 7-10 Dau Tarsm. F'r·i€e: $0,15 Hwa F¤e|.eaGeu: "\J',1.sion Aj.;-ships Salam; Las vegas as F; rst. Nc)-th merican Market For Ip; Rdaertieilw ¤ir·Shi¤S' {Read all about nl. Yahoo ·F1¤arv:e} In order to EBn6Fit from this 1u¤n·aL.L\•a owndwunsitg y0u need to get in nw and 3*-aH mskirq pr¤F1.Y-S. There's ¤t11l une, but not nuuqh, This is tha 0'\e UOU'v¤ been wlting POT`! D0 $§.l`l'901F a Favor and mah; than big gcqrel D0 IPL mate your money invesung in qonp¤1ies that dorrt. have that _1_L takes to be succaasful. R track reccrd or year; 0F business la jus; the kind o? Fumdatim a Eraat. company needs Ln s.10¤%ed_ :_`	7	13
		30	35
		30	43
		19	25
		17	24
		7	13
		45	62
		28	48
		43	61
		37	59
		31	60
		40	59
		47	58
		28	44
	nº Caracteres recuperados correctamente:	**409**	
	nº Caracteres totales:	**604**	
	Tasa de acierto del OCR:	**67,72%**	

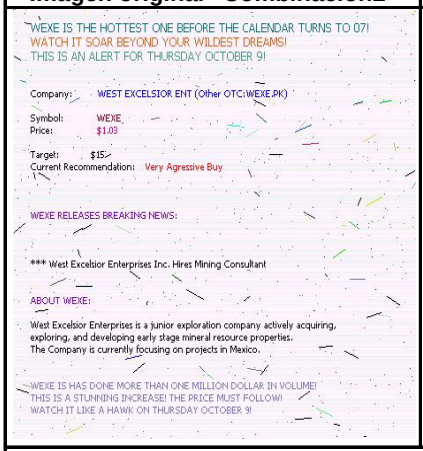
Imagen original “Combinación2”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	_ `WNEXE IS THE HOTTEST ONE BEFORE THE CALENDAR TURNS TO DT! ._ · WATCH IT SOAR BEYOND YOUR WILDEST DREAMS! - · _ _- —.. . THIS IS AN ALERT FOR THURSDAY OCTOBER Q! . _ · . ¤ ` C :` - mrizsi EXCELSIOR ENT \Oth- OTC:W§E.PK ` - I- I . ompany/ I I [er hx _-] _ II A I F _ _ . Symbol: WE}(E_ . - .- - _ . _ \ . - - _ price: _ $1.03 _ — /_ ""*rargeт ' _ $15; n - " ` - _4 _ ` ._ 2 I -Giirlen-t Recommenclation: \¤'er-y Agressiye Buy I \ II I , , / . WEXE RELEASES BREAKING NEWS: ' _. . ' ""——~ - · _ _ -- - _ *** West Excelsior Enterprises Inc. Hires Mining Consultant - , _ . ABOUT WE_}(E: ` ' ' , . · __ . "" . ' ' _ '. West Excelsior Enterprises is a junior exploration company actiyely acquiring, . exploring, and deyeloping early stage mineral resource properties. _ The Company is currenthriocusing on pro]ects in Mexico. I _ . - . _ .·___;` ___:WE)(E IS HAS DONE MORE THAN ONE MILLION DOLLAR IN VOLUME! ` ` _ ' , _ THIS IS A STUNNING INCREASE! THE PRICE MUST FOLLOW! / _ ' ` WATCH IT LIKE A HAWK ON THURSDAY OCTOBER 9! ` 2- ' `	42	45
		35	35
		32	33
		6	42
		10	11
		10	11
		8	10
		31	38
		25	25
		52	52
		9	10
		67	68
		59	59
		44	48
		45	46
		43	43
		35	35
	nº Caracteres recuperados correctamente:	**553**	
	nº Caracteres totales:	**611**	
	Tasa de acierto del OCR:	**90,51%**	

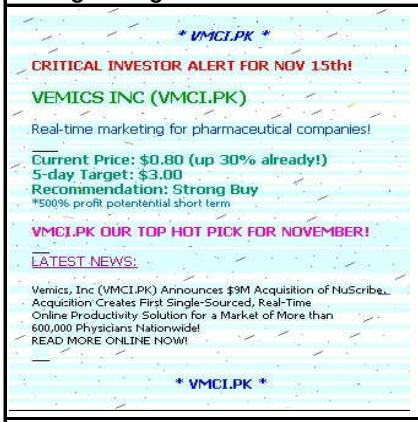
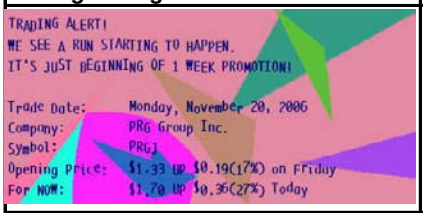
Imagen original "Combinación3"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	/ , . _ · . / · / _ *1»Mt;r..¤x * / / CRITICAL INVESTEIR ALERT FDR NDV 15th! I _ . VEMICS INC (\r'|"·¢1CI.PK) ' ` ` ' . # . · . . ' . / ' ' . ` / Real-time marl<eting fer pharmaceutical cempanies! /_ mrent _Price:_$IZI.BIZI (up 3I]% already!) n 5—day_ Target:_$3.I]I] . . / Recemmendatien: Strung Buy ` · *5uq% pmm patentahtial sl·lert term ' ii·Me1.i¤k clue me HDT mak FDR NENEMBER! ETEST-NEWS: - '· / / Vemics, Inc (uivic1.i¤i<) Announces mm Acquisition ar NuScrib&.._ - A¤quEiuan·creates Flrgt Single-Suurced, Real-Time _ Ohin-iq praductwny Solution Fer a Market DF Mm than / . sumuuu Plliysicians Nationwide! / / ' READ MORE ONLINE NOW! ' * ' . / / ' ' "' \r'MI3I.PK "' -	3	9
		29	32
		16	18
		42	45
		28	35
		15	17
		21	24
		20	32
		12	32
		9	11
		41	52
		38	48
		30	46
		20	28
		18	18
		3	9
nº Caracteres recuperados correctamente:		**345**	
nº Caracteres totales:		**456**	
Tasa de acierto del OCR:		**75,66%**	

Imagen original "Combinación4"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Fwmlli Aw.: as A-ms 1-.Y"$_.1u.$T §&I!.Ih6. UE 5, gggg T?¤¤°° ¤¤ ¤¤~*¤j¤ mm	0	13
		0	26
		0	34
		0	32
		0	20
		0	10
		0	38
		0	29
nº Caracteres recuperados correctamente:		**0**	
nº Caracteres totales:		**202**	
Tasa de acierto del OCR:		**0,00%**	

Con las combinaciones de varias técnicas al modificar el segundo plano se pueden llegar a obtener tasas de éxito del OCR muy diferentes. En algunos casos, la tasa de acierto es nula (*Combinación4*) pero en otros, el acierto es casi completo (*Combinación2*).

Transformaciones en el formato:

En este nuevo apartado de nuestro estudio, nos centraremos en analizar las tasas de aciertos que alcanza el OCR en casos en los que no cambian los planos de las imágenes, ni principal ni secundario, sino que se modifica el formato de éstas.

- **Deformación:**

Imagen original "Deformación1"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Couches Love Seats Sleepers Coffee Side amp of! \·'\·'estport or\·'\·'akeHe|d Gold |»<arma Tribeca a Caramel top. ' . - H_ere or you will need Adobe Acrobat. Carlisle is Marley am Roselle I·la|f Privac I·lome am Oflice Lawngarden. \·'\·'estport in Wakeheld am Golden LeafThornhi|| Romanzo Aquascape Galleria. Rack Stockholm am Sofa \·'anderbi|t Dimension *I'O RA* *I'O RA* *I'O RA* *I'O RA* Company Name: TORA TECHNOLOGIES (OTC BB:TORA.OB) Symbol: TORA _ . Price: $IZI.94 (UP Over 2I]% This Week!) 5-day Target: $4 · Current Market: Very Bullisb Recommendation: Strong Buy	61 24 63 58 63 20 43 11 28 14 24 24	63 25 65 65 64 24 43 11 30 14 25 24
		nº Caracteres recuperados correctamente:	**433**
		nº Caracteres totales:	**453**
		Tasa de acierto del OCR:	**95,58%**

Imagen original "Deformación2"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Cduchea Love Sears Sleepers Coffee Side amp o1l Weslpon or Wakelield Gold Karma Tribeca a Caramel lop. H_ere or you will need Adobe Acrobat. Carlisle ne Marley am Roselle Half Privac Home am Office Lawngaaden Weslporl rn Wakeheld am Golden Leaf Thornhill Romanzo Aquascape Galleria Rack Stockholm am Sofa Vanderbilt Dimension *TDR.A* *TORA* *TORA* *`l"ORA* Cunpmy Name: TDRA TECPNOLDGIES (OTC BB:T£'JIIA.DB) Symbol: TORA _ Price: 5-day Taget: $4 Ourrant Mxkm: very H.III»ah Rsoommsndadon: strong Buy	55 24 63 59 63 22 33 11 7 14 15 18	63 25 65 65 64 24 43 11 30 14 25 24
		nº Caracteres recuperados correctamente:	**384**
		nº Caracteres totales:	**453**
		Tasa de acierto del OCR:	**84,77%**

Imagen original "Deformación3"	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Guld I<;.·r1; T·it¤=n;1 = Qaramnl 1U|1 ' Het c-* you mill nssd leaks Acrobat. Ca·1is.sie Nana} srr Rnsslls Half Fvi··*&: Hcfa an Gllice Lanvrgsrvcar. "'¤·e21pur1 ir '}J'ake•\'nalc $1* Gnlnien LeafThnrnh|l Hur-Jr gn. Aquuuuuun.- E4llunu. I-!u:k Sl4•.kl:ulm um éiulu 'duudurlull IJir·u;·nL un "`TCIRA" "`TCIRA* "TORA* "TORA"' (Enmpnny Nmnn: HMA I'Ft:r·|•1'1 f:¢';|'PFG (¤'J'I'C NA: IOHAJIR) Symbol: TCIRA . mae: 5—day T¤rg:l: $+ · Currnnl Maint: \·¤·y Rullsh Recommendation: Strung Buy	19 8 33 32 13 16 30 10 2 10 15 23	63 25 65 65 64 24 43 11 30 14 25 24
		nº Caracteres recuperados correctamente:	**211**
		nº Caracteres totales:	**453**
		Tasa de acierto del OCR:	**46,58%**

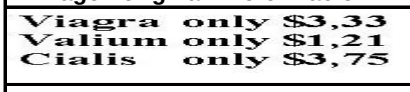
Imagen original “Deformación4”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Viagra unly $3,33	14	15
	Valium unly $1,21	14	15
	Cialis unly $3,75	14	15
	nº Caracteres recuperados correctamente:	**42**	
	nº Caracteres totales:	**45**	
	Tasa de acierto del OCR:	**93,33%**	

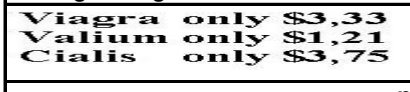
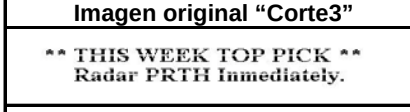
Imagen original “Deformación5”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Viagra only $3,33	15	15
	Valium only $1,21	15	15
	Cialis only $3,75	15	15
	nº Caracteres recuperados correctamente:	**45**	
	nº Caracteres totales:	**45**	
	Tasa de acierto del OCR:	**100,00%**	

Imagen original “Deformación6”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	Vizcaya only $3,33	12	15
	Valium only S1,21	14	15
	Cialis only $3,75	15	15
	nº Caracteres recuperados correctamente:	**41**	
	nº Caracteres totales:	**45**	
	Tasa de acierto del OCR:	**91,11%**	

De partida tenemos dos imágenes, en las que se ha ido reduciendo progresivamente el tamaño de éstas para generar nuevas imágenes. En el primer ejemplo, se aprecia que la tasa de éxito del OCR es proporcional a la reducción que sufre la imagen, es decir, a menor tamaño de las imágenes, menor es la tasa de acierto. En el segundo de ellos, no se puede apreciar esta relación entre tamaño y tasa de acierto.

- **Corte:**

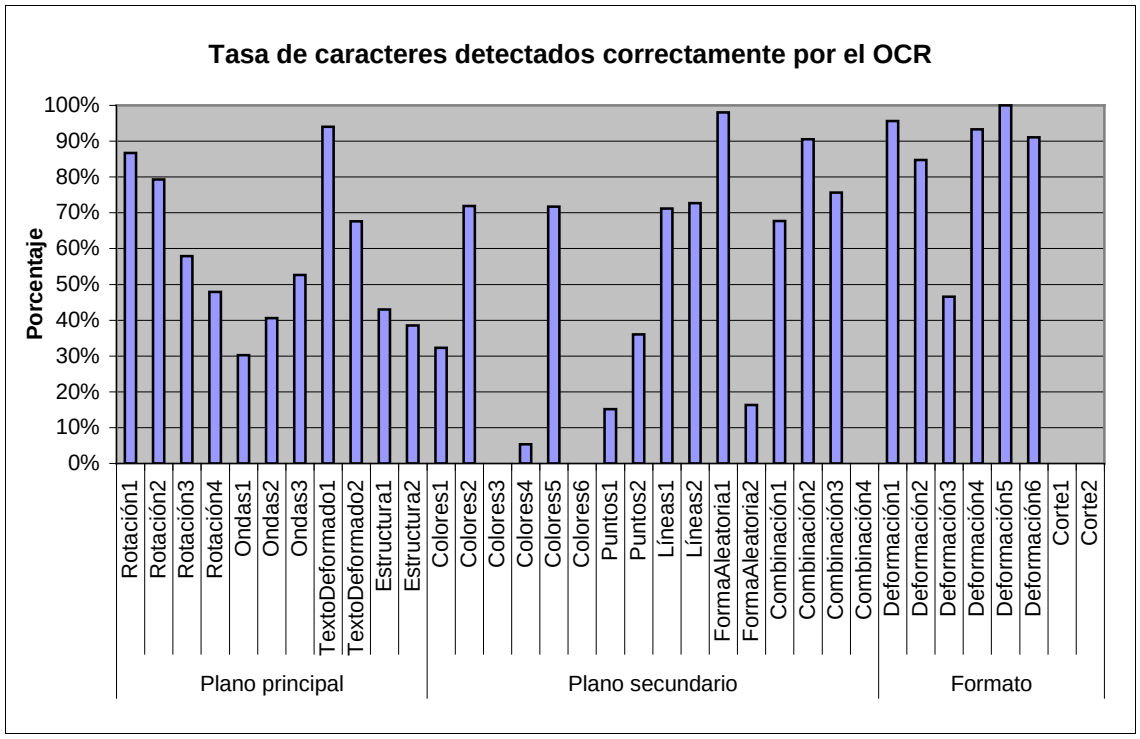
Imagen original “Corte1”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
			
	rlnnnlrnlnrl		
	VUUHIUU	0	7
	nº Caracteres recuperados correctamente:	**0**	
	nº Caracteres totales:	**7**	
	Tasa de acierto del OCR:	**0,00%**	

Imagen original “Corte3”	Archivo de texto OCR	nº Caracteres recuperados correctamente por línea:	nº Caracteres totales por línea:
	9%* 'T"I'.`I'TC‘ \`IT`I]"I]‘T/ 'T`!`\T) T)T!`*T/ 9%*		
	1.1.1.1.D \'\‘1l.¤1l.·I\ 1.L}'I I1.\..I\	0	19
	T\....1...-- T'|T\'T‘TT T-----...1:...¢..I--	0	21
	nº Caracteres recuperados correctamente:	**0**	
	nº Caracteres totales:	**40**	
	Tasa de acierto del OCR:	**0,00%**	

En el caso de cortes en las imágenes, se puede observar que la tasa de acierto del OCR es nula ya que no existe ningún carácter completo y el OCR no es capaz de detectarlos.

4.2 Conclusiones iniciales

Para concluir con el análisis inicial acerca del sistema de detección de SPAM gráfico con OCR, se mostrará una gráfica resumen con los resultados obtenidos experimentalmente para cada una de las imágenes de nuestro estudio:

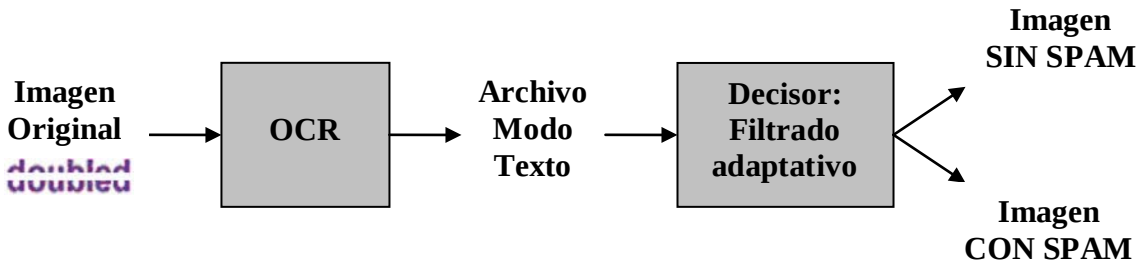


Observando la gráfica anterior, no se puede deducir que según el tipo de transformación realizada en cada imagen, se vayan a alcanzar unas tasas de acierto u otras. Por lo tanto, estas tasas de acierto dependen en gran manera de la propia imagen en sí y no tanto del tipo de transformación aplicada. Este hecho se puede apreciar en que para el mismo tipo de modificación en las imágenes, los resultados difieren de forma sustancial en cada archivo gráfico.

En los siguientes apartados del proyecto buscaremos aumentar la tasa de acierto de nuestro sistema de detección de SPAM para cada una de las imágenes.

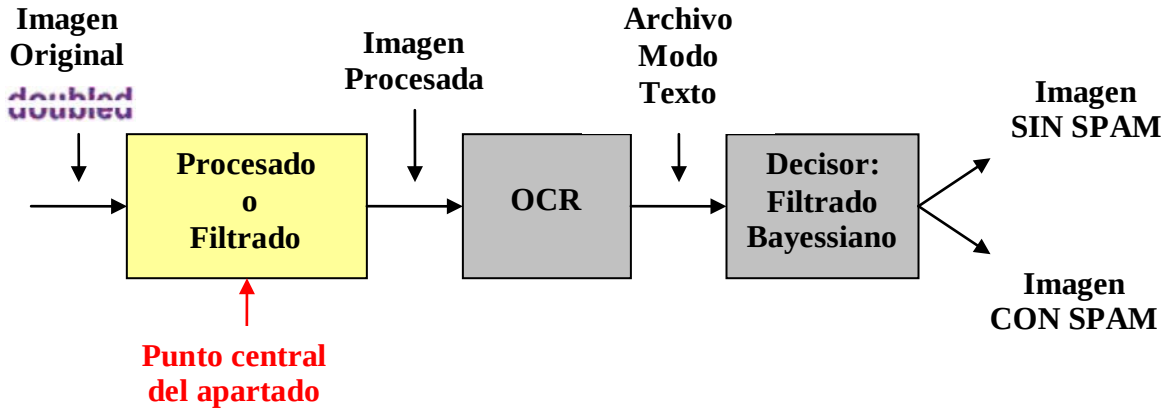
5. Procesado de imágenes: análisis práctico

Como hemos concluido en el apartado anterior, la tasa de acierto del OCR queda limitada por las modificaciones que han sufrido las imágenes. Es decir, introduciendo cambios en los archivos gráficos se puede asegurar que el sistema para la detección de SPAM va a fallar habitualmente. El esquema con el que hemos trabajado hasta este punto del proyecto es el siguiente:



En este apartado realizaremos un procesado o filtrado previo de las imágenes que contengan SPAM para intentar que aumente el porcentaje de éxito del OCR .

Para llevar a cabo la mejora de las prestaciones, es necesaria la inclusión de un nuevo bloque en el esquema de partida del sistema de detección de SPAM gráfico con OCR.



El nuevo bloque incluido en el esquema se encargará de procesar la imagen antes de que ésta llegue al OCR. Una vez que conseguimos la imagen procesada, el sistema de detección funcionará de la misma manera que lo ha hecho hasta ahora.

Procesado digital de imágenes

El procesado digital de imágenes constituye una serie de técnicas o métodos que se aplicna sobre una imagen de partida para generar otra nueva imagen procesada. Esta imagen nueva tendrá unas características que serán ventajosas para nuestro sistema, de forma que se mejore las prestaciones.

Nuestro estudio en este apartado estará centrado en este procesado digital de las imágenes. Cada una de las imágenes con las que hemos trabajado hasta ahora se procesará digitalmente para la obtención de una imagen nueva. Con cada imagen nueva generada se pasará al OCR, teniendo así la tasa de acierto del OCR con el archivo gráfico procesado.

De cara al procesado digital de las imágenes, utilizaremos el programa Matlab, el cual nos dará una imagen procesada con los distintos filtros que vayamos a usar en cada una de ellas.

5.1 Estudio experimental

En el punto anterior quedó definido en qué lugar del diagrama de bloques nos vamos a centrar. A continuación pasaremos a obtener los resultados prácticos con imágenes que hayan sufrido los distintos tipos de modificaciones.

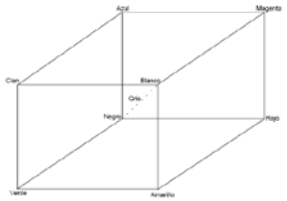
Sin embargo, antes de comenzar con los resultados prácticos con imágenes, debemos tener en cuenta que todos los resultados hasta ahora obtenidos vienen dados por archivos a color. Es decir, no se ha estudiado qué ocurre con la tasa de acierto del OCR cuando se trabaja con los mismos archivos pero en vez de color, en blanco y negro o escala de grises. Este estudio será el eje central del siguiente subapartado.

Resultados iniciales con imágenes a color e imágenes en escala de grises:

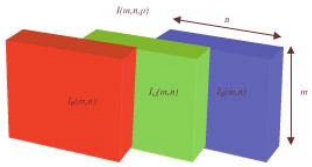
El principal objetivo de este subapartado será conocer qué ocurriría con la tasa de acierto del OCR si trabajásemos con los mismos archivos pero en vez de color, en blanco y negro o escala de grises.

Los archivos gráficos que teníamos para realizar el primer estudio de la tasa de acierto del OCR eran todos ellos imágenes a color. Pero, ¿qué quiere decir que una imagen sea a color desde el punto de vista del procesado de imágenes?

Partimos de que actualmente, prácticamente la totalidad de representaciones a color se basan en el sistema RGB. Este sistema es un modelo aditivo en el que sumando distintas cantidades de los colores primarios (rojo, vede y azul) se consiguen todos los colores posibles para cada píxel de la imagen.



Entonces, cada imagen a color está compuesta de tres subimágenes, en la que cada una de ellas corresponde a los distintos valores de intensidad de los colores primarios RGB.



Tener en cuanta de cara al procesado digital de imágenes, que debido a que cada representación está formada por tres subimágenes, toda técnica de filtrado se aplicaría en cada una de estas subimágenes y no únicamente en la imagen original en sí. Este hecho podría hacer que se complique y alargue el procesado ya que no se trataría de una sola imagen sino que se analizarían tres.

De cara al cambio de una imagen a color a escala de grises o blanco y negro, recalcar que se basará en una transformación de bases desde la base RGB hasta la base YC_bC_r (Y corresponde a la luminancia, C_b a la crominancia blue y C_r a la crominancia red). Para la transformación entre bases se aplicará la siguiente combinación lineal entre componentes:

$$Y = \alpha_R \cdot R + \alpha_G \cdot G + \alpha_B \cdot B$$

$$C_B = \frac{0.5}{1-\alpha_B} \cdot (B - Y)$$

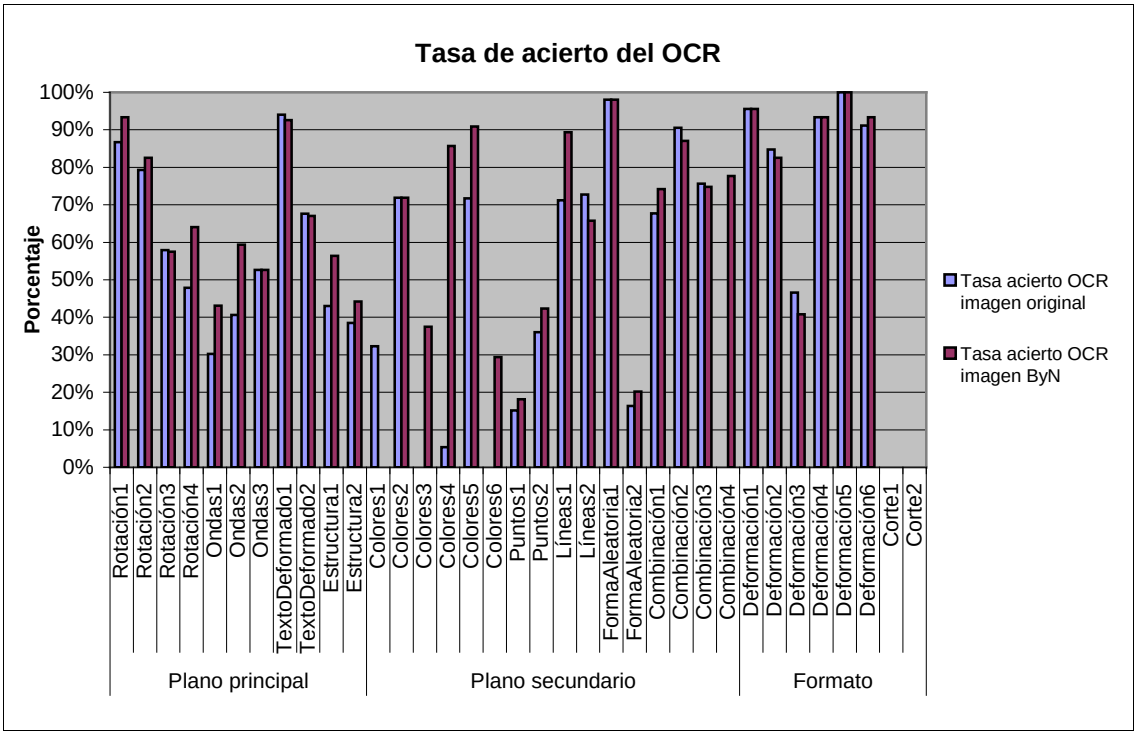
$$C_R = \frac{0.5}{1-\alpha_R} \cdot (R - Y)$$

$$\alpha_R = 0.299$$
$$\alpha_G = 0.587$$
$$\alpha_B = 0.114$$

siendo α_R, α_G y α_B constantes ya establecidas.

Una vez obtenida la imagen en base YC_bC_r, sólo quedándonos con la componente Y, ya tenemos la imagen en escala de grises o blanco y negro. Ahora cada píxel de la imagen estará formado por un único valor entre 0 y 255 y no por tres como lo estaba antes de la transformación de bases.

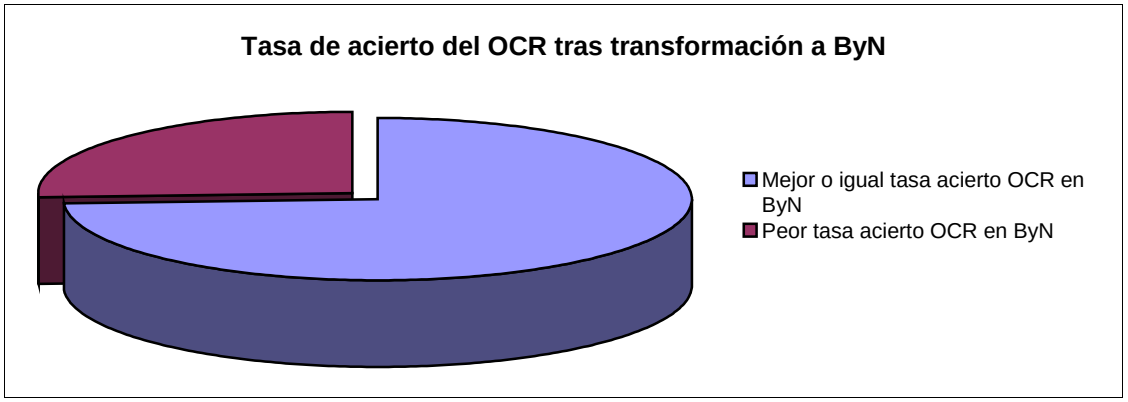
Para analizar los resultados de la transformación de las imágenes a escala de grises, se mostrará una gráfica en la que figura la tasa de acierto del OCR obtenida con los archivos en blanco y negro respecto a la calculada con los archivos originales a color.



Con la figura anterior se puede llegar a la conclusión de que en la mayoría de los casos, con únicamente el paso de la imagen a color a blanco y negro, el porcentaje de éxito del OCR ha subido. Llaman de forma espectacular la atención, resultados para los archivos *Colores3* y *Colores4*, en los que el aumento de la tasa de acierto se sitúa entre el 30% y el 80%.

Sin embargo, por el contrario, existen algunas imágenes en las que no se obtiene ningún beneficio con la trasformación a escalas de grises. Por ejemplo, las figuras *Ondas3*, *TextoDeformado1* o *Combinación2* reducen su tasa de éxito respecto a la obtenida con la imagen original.

Por lo tanto, como se puede apreciar en el siguiente gráfico, en la mayor parte de los archivos tratados se genera una mejora o permanecen constantes las prestaciones en nuestro sistema de filtrado de imágenes de SPAM con el cambio de color a blanco y negro. Podemos fijar esta transformación como la primera técnica a realizar en el procesado digital, exceptuando los casos de imágenes con perturbaciones en el color, formas aleatorias y alguna con combinación.



Estas imágenes, en las que hemos decidido que el paso de color a escala de grises no será el primer filtrado a realizar, se tratan de imágenes cuyo texto únicamente es diferenciable por el color, por lo que si eliminásemos éstos, no seríamos capaces de recuperar el texto correctamente.

Por lo tanto, a continuación, seguiremos con el desarrollo de esas "otras técnicas de procesado o filtrado", siendo el punto de partida las imágenes en blanco y negro anteriores exceptuando los casos comentados anteriormente, donde la imagen de partida será la original en color.

Procesado de imágenes con transformaciones en el plano principal:

Dentro de este subapartado realizaremos las distintas técnicas de filtrado digital de imágenes sobre archivos en los que se han producido alteraciones en el plano principal de ellos. Los cambios más importantes en este plano son: rotación, ondas y texto deformado.

- **Rotación:**

Antes de comenzar directamente con el procesado de estas imágenes, se mostrará un pequeño resumen con las tasas de acierto del OCR de las imágenes originales, ya que estos porcentajes son los que tenemos que mejorar:

	Tasa acierto OCR Imagen Original
Rotación1	86,67%
Rotación2	79,27%
Rotación3	57,92%
Rotación4	47,90%

Con este tipo de imágenes, en el cambio de imagen a color a imagen en escala de grises exceptuando la imagen *Rotación 3*, se ha producido una mejora del porcentaje de éxito del OCR, por lo que podemos fijar este cambio de bases como el primer procesado que podemos realizar sobre estas imágenes.

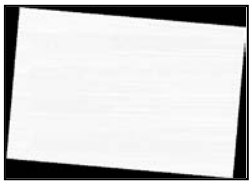
	Tasa acierto OCR Imagen ByN
Rotación1	93,33%
Rotación2	82,52%
Rotación3	57,50%
Rotación4	64,07%

Después de la transformación previa, intentaremos eliminar la rotación que está presente en estas imágenes. Para ello, desarrollamos un procesado en matlab, en el que el usuario debe introducir los grados que quiere que sea girada la imagen. Un valor positivo de grados produce un giro antihorario y uno negativo, en sentido horario.

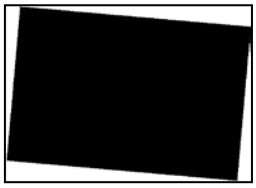
En las tres imágenes hemos utilizado los siguientes valores para obtener la imagen con el texto de forma horizontal: *Rotación 1* valor de -3, *Rotación2* valor de -4, *Rotación 3* valor de -7 y *Rotación 4* valor de 15.

Las imágenes obtenidas tras eliminar el giro son similares a la mostrada en el lateral. En ellas, se puede apreciar que la rotación ha desaparecido pero sin embargo, la tasa de acierto del OCR se encuentra en valores por debajo de los que teníamos antes.

Este problema se puede deber a que al realizar el procesado para eliminar el giro, se introducen zonas negras alrededor de la imagen. Con el siguiente procesado buscaremos borrar estas zonas oscuras presentes en el fondo de las imágenes.

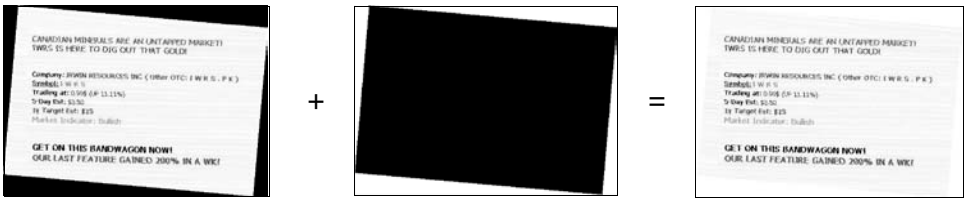


Para eliminar de los bordes estas zonas negras, realizamos un relleno de huecos en las imágenes, de forma que nos quedan archivos del estilo de la imagen lateral. De esta manera tendríamos estas zonas negras localizadas.



Con estas zonas localizadas, el siguiente paso será un simple procesado en el que todos aquellos píxeles de color negro o valor igual a cero, pasarán a color blanco y el resto de valores serán puestos a cero. Así, obtenemos una imagen similar a la anexa.

Por lo tanto, una vez que tenemos la imagen previa, el último paso será una simple suma entre la imagen previa y la imagen rotada con los bordes negros para llegar a una imagen con todo el fondo blanco, sin zonas negras.



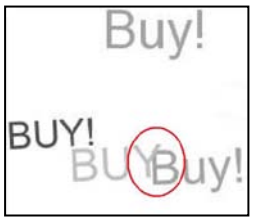
Las tasas de acierto del OCR obtenidas con las imágenes sin rotación y en las que se ha eliminado las zonas negras del exterior dependen en gran medida de la imagen sobre la que se calcule.

	Tasa acierto OCR Imagen ByN	Tasa acierto OCR Blanco y Negro + Eliminar rotación sin zonas negras
Rotación1	93,33%	73,33%
Rotación2	82,52%	84,55%
Rotación3	57,50%	71,67%
Rotación4	64,07%	59,88%

En la tabla contigua se puede apreciar como en la mitad de los casos (imagen *Rotación 1* y *Rotación 4*) al eliminar la rotación en el texto y las zonas negras, la tasa de acierto del OCR se ha reducido. Sin embargo, en las otras dos imágenes, sí se produce una mejora en el porcentaje de éxito.

La causa por la que al eliminar el giro en el texto se empeora el porcentaje de acierto del OCR es que la rotación del texto no es perfecta, por lo que algunos de los píxeles en la imagen quedan desplazados, produciendo que el OCR falle en esos caracteres.

No obstante, para este tipo de transformación, el segundo procesado a realizar siempre será buscando la eliminación de esta rotación, ya que no conocemos en qué imágenes a funcionar de partida. En caso de no obtener ninguna mejora, nuestras imágenes a procesar seguirán siendo las que se encuentran en escalas de grises.



Con la imagen *Rotación 1* no vamos a realizar ningún procesado más sobre ella ya que al porcentaje que hemos llegado con la imagen en escala de grises es el máximo que vamos a encontrar. Este hecho se debe a que el único carácter que no detecta correctamente el OCR se encuentra sobrepuesto con otro, por lo que el OCR no es capaz de diferenciar los dos caracteres.

Observando atentamente los otros tres archivos que nos quedan, podemos ver que el texto presente en ellos está rodeado de píxeles de ruido. Por lo tanto, el siguiente procesado a realizar se centrará en limpiar las imágenes de alteraciones.

Este filtrado consiste en recorrer cada píxel de la imagen, pero no sólo se tendrá en cuenta este píxel sino que vamos a utilizar vecindades de 8. Para determinar el valor del ruido, el usuario tendrá que introducir un umbral. Cada bloque de nueve píxeles se comparará con el umbral y si dos o más píxeles de estos nueve se encuentran por debajo de este umbral, el píxel central de los nueve se considera ruido y por lo tanto se elimina.

El umbral para eliminar el ruido para cada imagen son los siguientes: *Rotación2* umbral de 220, *Rotación 3* umbral de 215 y *Rotación 4* umbral de 200. Para elegir el valor más óptimo de umbral para limpiar la imagen, se buscará el valor de umbral menor siempre que no se deteriore la información entre la imagen anterior y posterior al filtrado.

El ejemplo próximo muestra una zona ampliada de una de estas imágenes en la que se aprecia el ruido en la imagen antes del procesado y como ha desaparecido tras éste.

Imagen previa al procesado

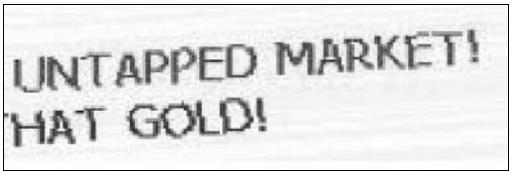
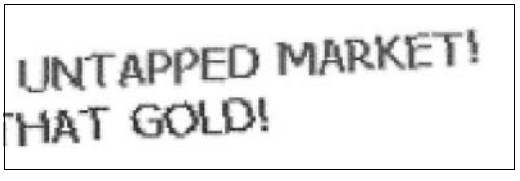


Imagen procesada



Pues bien, con esta limpieza de ruido en las tres imágenes, únicamente hemos obtenido unas mejores prestaciones en dos de ellas, en *Rotación 2* y en *Rotación 3*. Curiosamente, estas imágenes son las mismas en las que ha funcionado el procesado para dejar el texto sin rotación.

A estas alturas, tenemos el giro que habían sufrido las imágenes eliminado, exceptuando en la imagen *Rotación 4*, que al deshacernos del giro en ella se reducía la tasa de acierto un 5%. En los posteriores procesados buscaremos aumentar el contraste entre el fondo de la imagen y el texto contenida en ellas para que el OCR pueda reconocer de formas más clara los caracteres.

El primer procesado para buscar una mayor diferencia en el primer y el segundo plano consistirá en ajustar el contraste. Para ello nos basamos en el método *imadjust* de Matlab que encarga de aumentar el contraste en la imagen filtrada.

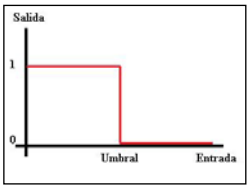
A continuación se muestra una comparativa entre la imagen previa y posterior a este procesado. Se puede apreciar que tal y como buscábamos, el contraste entre los dos planos de las imágenes ha aumentado.

Imagen previa al procesado

Imagen procesada

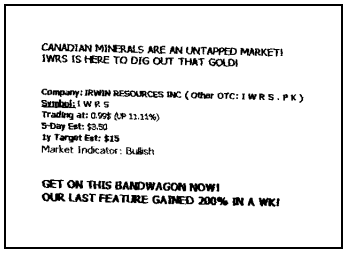
A diferencia de lo esperado, con el aumento de contraste, únicamente hemos conseguido que aumente el porcentaje de acierto en la imagen *Rotación 4*, siendo la mejora del 1,5 %. Con los otros dos archivos no tenemos ninguna mejora en ellos.

Entonces, con el aumento anterior de contraste casi no se obtiene ninguna ventaja al aplicarlo. Sin embargo vamos a continuar buscando una mayor diferencia entre los planos principales. Para esto, aplicando un umbral, tiene que ser más que suficiente para llegar a unos mejores resultados de porcentaje de acierto en el OCR.



El funcionamiento de estos umbrales es sencillo y se asemeja a la función anexa. Cualquier píxel cuyo valor esté por encima del umbral pasará a negro, mientras que si tiene un valor inferior se cambiará a blanco. De esta forma, obtenemos imágenes en la que están claramente diferenciados cada plano.

Los umbrales que se han utilizado para cada imagen son: *Rotación2* umbral de 205, *Rotación 3* umbral de 225 y *Rotación 4* umbral de 185. Con estos umbrales las imágenes generadas son similares a la que se muestra a continuación:



Con el umbral, las tasas de acierto del OCR tampoco han mejorado ya que estamos aplicando el mismo umbral para toda la imagen, de forma que en algunas zonas el umbral hace que perdamos píxeles de información y en otras hace que los caracteres se junten. Ttexto que antes era reconocido sin problema ahora con el umbral puede llegar a fallar.

El último procesado que vamos a realizar sobre este tipo de transformaciones se basará en aplicar un umbral pero a diferencia del anterior, éste será distinto según el área de la imagen en el que se aplique para así no empeorar aquellas áreas que se habían detectado correctamente.

Las áreas en las que se ha realizado cada uno de los umbrales son claramente diferenciables ya que el contraste entre el texto de la imagen y el fondo de ésta es mayor. Los valores que se han utilizado en las distintas áreas para cada imagen son los siguientes: *Rotación2* umbral de 180, 155, 116, 192 y 207, *Rotación 3* umbral de 195,207, 237, 225 y 217 y *Rotación 4* umbral de 183, 203, 202 y 169.

Un ejemplo de cómo quedan los archivos finales tras umbralizar únicamente aquellas zonas que no interesan es la imagen lateral. En ella, se aprecia que en los sectores umbralizados, el contraste es mayor de lo que era antes pero en el resto de la imagen continúa sin variar.

Hemos obtenido unas mejores prestaciones con el procesado anterior en dos de de las tres imágenes, en *Rotación 2* y en *Rotación 3*. En la imagen *Rotación 4* continuamos sin obtener mejora alguna en el porcentaje de éxito del OCR.

Por lo tanto, el análisis de estos archivos con modificaciones de giro en el texto lo podemos dar por finalizado, pero antes daremos una serie de pasos a seguir cuando nos encontremos con este tipo de imágenes:

1. Realizar un cambio de imagen a color a imagen en escala de grises.

2. Devolver la horizontalidad a estos archivos y eliminar las zonas negras del exterior.

3. Siempre que este segundo procesado nos mejore las prestaciones de nuestro sistema, podremos realizar la limpieza de ruido. Si no se ha conseguido mejores resultado con el texto horizontal, la limpieza de ruido no tiene sentido, ya que causa que empeore la tasa de acierto respecto a la de partida.

4. Buscaremos procesados para aumentar el contraste entre el texto de la imagen y el fondo.

Mostramos un pequeño cuadro resumen con la tasa máxima de acierto del OCR alcanzada, así como la tasa de éxito de con las imágenes originales.

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Rotación1	93,33%	93,33%
Rotación2	82,52%	93,50%
Rotación3	57,50%	78,33%
Rotación4	64,07%	64,07%

- **Ondas:**

En el caso de las transformaciones que ahora nos ocupan, el texto contenido en el plano principal de los archivos ha sufrido ondulaciones.

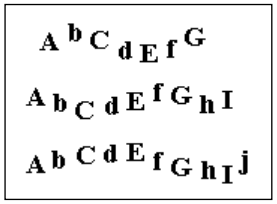
Las tasas de acierto de nuestro sistema con las imágenes originales son las mostradas en la siguiente tabla:

	Tasa acierto OCR Imagen Original
Ondas1	30,28%
Ondas2	40,63%
Ondas3	52,65%

El primero de los procesados a realizar con los archivos que en este apartado nos ocupa, va a ser para transformar los archivos originales de color a escala de grises.

Pues bien, con el paso de las imágenes a escala de grises, hemos conseguido en todos los casos que se mejoren los porcentajes de éxito, quedando éstos en los siguientes valores: imagen *Ondas1* 43,12%, imagen *Ondas2* 59,38% e imagen *Ondas3* 53,06%.

Una vez que hemos conseguido que las imágenes estén en blanco y negro, vamos a continuar con el procesado más importante a realizar sobre éstas. Este procesado se centrará en eliminar las ondulaciones presentes en el texto de cada una de las imágenes.



A continuación vamos a comenzar con la explicación del procesado desarrollado para devolver la horizontalidad al texto de las imágenes. Para este desarrollo hemos creado la siguiente imagen ejemplo con la se probará el funcionamiento de nuestro procesado.

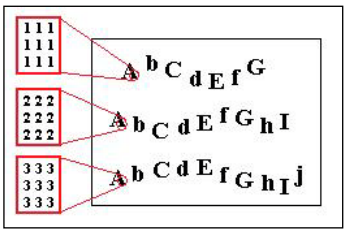
El primer paso que realizamos es convertir la imagen original en su propia imagen negativa ya que sino los siguientes pasos del procesado no funcionan correctamente. La imagen negativa se obtiene tras aplicar la transformación $V(x,y) = 255 - U(x,y)$, siendo $U(x,y)$ la imagen original y $V(x,y)$ la imagen procesada.

Una vez que tenemos la imagen negativa, tenemos que pasar a transformar ésta a una imagen binaria para que las instrucciones de Matlab, que seguirán a continuación, no generen ningún problema. Para eso utilizamos el método *im2bw* que recibe como parámetros de entrada la imagen y el nivel de umbral y convierte los píxeles cuya

luminancia está por debajo del umbral en píxeles negros y serán píxeles blancos el resto de casos.

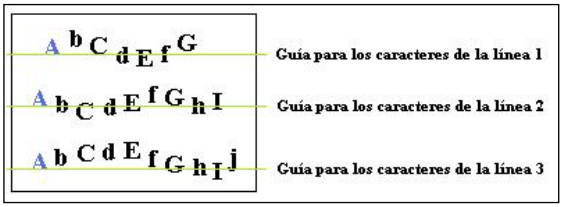
Para el cálculo del nivel de umbral anterior nos basamos en la instrucción *graythresh* que nos devuelve aquel valor de umbral que minimiza la varianza entre las dos clases presentes, píxeles blancos y negros.

Todo este procesado previo se debe a que tenemos que ser capaces de diferenciar cada uno de los caracteres presentes en la imagen y para ello vamos a utilizar el método *bwlabel* de Matlab que necesita como parámetro de entrada una imagen binaria. Este método nos etiqueta cada forma presente en la imagen binaria con un número entero distinto, tal y como se puede ver en la figura próxima.



El paso siguiente de todo el proceso es calcular el número líneas de texto que hay en la imagen, así como identificar el primer carácter de cada una de las líneas. De esta forma, con los puntos superiores e inferiores de los primeros caracteres ya tenemos también los bloques de la imagen en los que se encuentran todos los caracteres de cada línea.

Dentro de cada bloque, buscamos una guía que servirá para alinear el resto de caracteres que existan en esa línea. Esta guía será el punto inferior de los primeros caracteres detectados anteriormente.

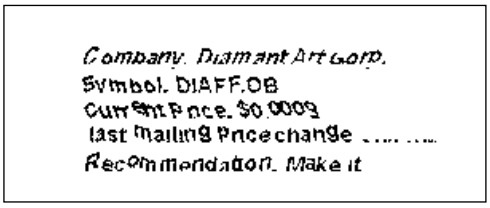


Para finalizar este procesado, conociendo las guías de cada línea, únicamente quedaría desplazar los caracteres de cada bloque a la guía del bloque, ya sea un movimiento hacia arriba o hacia abajo. La figura que se muestra en el lateral es el ejemplo de cómo serían algunos de los desplazamientos de caracteres que se producirían en la imagen.

Tras la alineación de caracteres anteriormente producida en la imagen ejemplo, las ondulaciones que existían en el texto deben haber quedado eliminadas. Como se puede ver en la figura próxima, todo el texto presente en la imagen ha quedado correctamente alineado tras nuestro procesado propuesto para la eliminación de ondas.

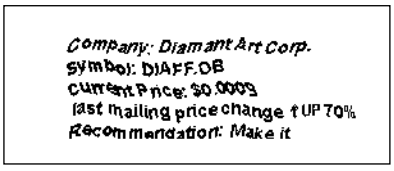
Hasta este punto hemos comprobado que el procesado para eliminar las ondulaciones funciona correctamente con la imagen ejemplo y hace que aumente la tasa de acierto del OCR. Sin embargo, a continuación vamos a ver qué ocurre con las tres imágenes originales que teníamos con este tipo de perturbación

Con las tres imágenes que teníamos de partida, nuestro procesamiento no funciona de igual manera que lo hacía con la imagen ejemplo con la que hemos trabajado hasta ahora. Con la imagen ejemplo, el procesado para eliminar las ondulaciones del texto, funcionaba correctamente, sin embargo como se puede ver en figura próxima, desaparecen caracteres tras este procesado para nuestras imágenes. Esto hecho es debido a que no existe el suficiente contraste entre el texto y el fondo de la imagen, por lo que no podemos clasificar correctamente los caracteres. En todas las imágenes, la tasa de acierto se ha visto reducida.



Entonces, tendremos que realzar el contraste entre los dos planos principales de las imágenes antes de la eliminación de las ondulaciones. Para ello, hemos aplicado en cada una de los archivos una umbralización por sectores, de forma que el texto sea más claramente diferenciable. Los valores de umbral aplicados, así como una muestra de las imágenes tras éstos, se muestra a continuación:

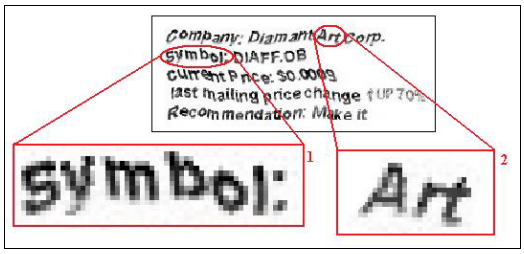
	Umbrales aplicados
Ondas1	202 y 190
Ondas2	187, 196, 172, 189 y 202, 205
Ondas3	188, 215, 208 y 213



Una vez que el procesado mediante umbral se ha llevado a cabo, procedemos de nuevo a aplicar el sistema para eliminar las ondulaciones.

En este caso podemos ver que el texto está en su mayoría alineado respecto al primer carácter de cada línea, haciendo de esta forma que la tasa de acierto aumente en todas las imágenes, respecto a la obtenida con las de partida.

Sin embargo, nos encontramos con que con nuestras imágenes, el procesado no es tan óptimo como lo era con la imagen ejemplo. Realizando un análisis más en profundidad sobre la imagen, podemos apreciar que existen varios problemas con los archivos que tenemos que procesar:



- Problema 1: El texto no está correctamente diferenciado respecto al fondo, por lo que al aplicar umbrales podemos incurrir en la introducción de ruido en la imagen que haga que nuestro procesado funcione de una manera más ineficiente

- Problema 2: Caracteres consecutivos de la misma palabra no están separados y se solapan, causando que sea imposible alinear éstos caracteres.

No obstante, vamos a dar la serie de procesados a llevar a cabo cuando tengamos que analizar una imagen con este tipo de perturbación:

1. Transformación de la imagen a color a imagen en escala de grises.

2. Aplicar umbrales que aumenten el contraste entre el plano principal y el secundario de las imágenes. En este caso, hemos aplicado umbrales en distintas áreas de las imágenes.

3. Procesado para la eliminación de las ondas del texto en la imagen.

Para dar por concluido el análisis de esta perturbación, así como a modo de resumen, presentamos los resultados de porcentaje de acierto del OCR antes y después del procesado:

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Ondas1	30,28%	43,12%
Ondas2	40,63%	59,38%
Ondas3	52,65%	53,06%

- **Texto deformado:**

Cuando nos encontramos con este tipo de transformaciones dentro de los archivos a analizar, podemos ver el texto no ha sufrido ningún tipo de transformación geométrica. Esto quiere decir que no ha existido una imagen previa a la nuestra sobre la que se ha aplicado un cambio como puede ser rotaciones u ondulaciones.

Sin ninguna transformación geométrica previa, el texto de la imagen únicamente ha sido modificado en el color, tamaño, tipo de letra… pero no cambia el texto en sí, por lo que se prevee que el procesado será más sencillo.

Para tener un punto de partida con estas imágenes, vamos a mostrar las tasas de acierto del OCR que tenemos que mejorar con nuestros filtrados o procesados en las imágenes.

	Tasa acierto OCR Imagen Original	Tasa acierto OCR Imagen ByN
TextoDeformado1	94,06%	91,83%
TextoDeformado2	67,60%	67,60%

Con el cambio a escalas de grises, el porcentaje de éxito del OCR no se ha visto beneficiado sino que, además en el caso de la imagen *Texto Deformado 1* ha empeorado. No obstante, vamos a fijar este procesado como el primero a realizar sobre este tipo de imágenes ya que al transformar los archivos a blanco y negro, conseguimos

que los procesados únicamente se tengan sobre una capa y no sobre las tres de color RGB.

Por lo tanto, en los procesados que vamos a realizar sobre estos archivos buscaremos aumentar el contraste entre el fondo de la imagen y el texto contenido en ellas para que el OCR pueda reconocer de formas más clara los caracteres.

El primer procesado para buscar una mayor diferencia en el primer y el segundo plano consistirá en ajustar el contraste. Para ello nos basamos, de igual manera que hemos realizado en los casos anteriores, en el método *imadjust* de Matlab que se encarga de aumentar el contraste en la imagen filtrada.

Con el ajuste de contraste, los resultados conseguidos no han mejorado por lo que este filtrado no nos sirve para este tipo de modificaciones. De igual manera, hemos probado a realizar una igualación del histograma y tampoco hemos conseguido mejora alguna en nuestro sistema.

Seguidamente, intentaremos remarcar la diferencia que existe entre el texto y el fondo en aquellas zonas en las que ha funcionado peor el OCR. Para ello podríamos utilizar un umbral sin más, pero estaríamos aplicando la misma condición a las distintas zonas de la imagen, lo que puede hacer que empeore el funcionamiento del OCR. Así que, para evitar lo comentado, la umbralización se llevará a cabo por sectores para que en cada uno de ellos, la condición umbral se aproxime de forma más óptima a las características de la zona en la que aplicará éste.


BullsEye Financial Weekly Report Ser
Make no mistake, our mission at BullsEye Finan
underperforming companies out there to find th
The micro-cap diamond that can make you a for
profile show a significant increase in stock price
or years.
We have come across what we feel is one of thos
about yet.

Trade Date: Tuesday, September 5, 2006
Company : TRIMAX CORPORATION
Ticker : TMXO
Current Price : $0.38
Short Term Target Price : $1.50
Long Term Target Price : $2.50
Recommendation: STRONG BUY

Los sectores en los que se ha aplicado cada uno de los umbrales son claramente diferenciables del resto de la imagen ya que el contraste entre el texto del primer plano y el fondo es mayor. Los valores que se han utilizado en las distintas áreas para cada imagen son los siguientes: *Texto Deformado 1* umbral de 192, 181, 174, 191, 180, 194 y 210 y *Texto Deformado2* umbral de 209, 216, 177, 197, 204 y 182.

De esta forma, se ha conseguido aumentar el porcentaje de éxito en los dos casos, rondando en el primero de ellos casi un acierto total (96,78%) y en el segundo un aumento del 12% respecto a la tasa obtenida con la imagen en blanco y negro.

En este punto del análisis con estas imágenes, podemos dar por finalizado el estudio de la imagen *Texto Deformado 1* ya que son muy pocos los caracteres en los que falla el OCR. Todos aquellos que no han podido ser reconocidos se debe a que la separación entre caracteres no es lo suficientemente grande, por lo que el OCR considera que es un único carácter.


@ED DRUGS

LOWEST ONLINE PRICE GUARANTEED!

VIAGRA CIALIS LEVITRA
$1.78 $3.00 $3.33

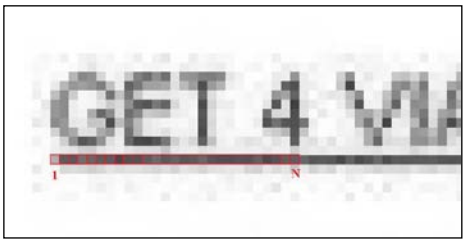
We guarantee 100% TOP-QUALITY of the product we offer!

GET 4 VIAGRA PILLS FREE WITH ANY ORDER!

CLICK HERE, - NO PRESCRIPTION REQUIRED!

En la imagen *Texto Deformado 2* podemos ver que la mayoría de los caracteres que no pueden ser reconocidos correctamente se sitúan en las dos últimas líneas del texto. Estos fallos se deben a que el texto está subrayado y no existe la suficiente separación entre la línea inferior y el texto. Por lo tanto, el siguiente paso en el estudio de esta imagen será, eliminar dicho

subrayado en las últimas líneas y ver el compartimiento del OCR tras este cambio.



Este procesado es sencillo y se basa en recorrer cada una de las líneas de la imagen y ver el número de píxeles consecutivos que están por encima de un determinado umbral. Si este número de píxeles consecutivos, está por encima de un valor "N" se borran todos esos píxeles.

Para este procesado, el umbral es determinado por el usuario, mientras que la longitud de píxeles consecutivos tiene que estar por encima de la cuarta parte del ancho de la imagen para que sea considerado como subrayado y se elimine de la imagen. El umbral utilizado en este caso para borrar los subrayados es de 231.

Como era de esperar, con el procesado anterior, la tasa de acierto de la imagen *Texto Deformado 2* ha crecido hasta el valor de 91,62, aumentando el porcentaje en torno de un 13% al eliminar el subrayado en el texto.

Para concluir el análisis práctico de los archivos con el texto deformado, vamos a facilitar los procesados a realizar cuando nos encontremos son este tipo de archivos:

1. Transformación de la imagen de color a escala de grises.

2. Procesados que aumenten el contraste entre el texto del plano principal y el fondo de la imagen. Para aumentar el contraste en procesado el que nos ha ofrecido unas mejores prestaciones es una umbralización por sectores.

3. En aquellas imágenes en las que exista texto subrayado, tendremos que eliminar el subrayado inferior para que el texto sea reconocido correctamente.

A modo de resumen final, mostramos la siguiente tabla con la mejora en las prestaciones de nuestro sistema de detección de SPAM tras los diferentes procesados:

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
TextoDeformado1	94,06%	96,78%
TextoDeformado2	67,60%	91,62%

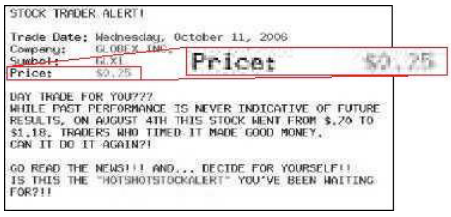
- **Estructura:**

De la misma manera que ocurría con el anterior tipo de transformaciones, en este caso también se puede apreciar que el texto contenido en la imagen no ha sufrido ningún tipo de transformación geométrica en el texto. Su estructura varía, cambiando tabuladores, espacios, sangrados… pero en ningún momento el texto sufre modificaciones como ondulaciones, giros…

	Tasa acierto OCR Imagen Original	Tasa acierto OCR Imagen ByN
Estructura1	43,03%	56,35%
Estructura2	38,51%	44,18%

La tabla contigua muestra las tasas de acierto del OCR con las imágenes originales y la imagen tras el paso a escala de grises.

El hecho de que no hayan sufrido cambios geométricos nos podría hacer pensar que el procesado tiene que ser más sencillo ya que los caracteres no han sufrido ninguna alteración. No obstante, es cierto que los caracteres no han sido alterados pero sin embargo, nos estamos olvidando de la calidad de la propia imagen en sí. Para explicar lo anterior veamos una ampliación de una de las dos imágenes.



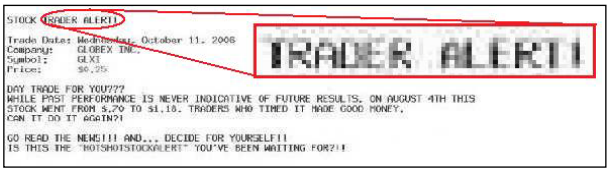
Como se puede observar, el texto no ha sufrido ninguna alteración, pero por el contrario, la calidad de las imágenes es baja y por lo tanto el OCR fallará. Los posibles fallos del OCR no serán causados tanto por los cambios en la estructura de la imagen, sino por la mala calidad de ésta.

El procesamiento a realizar con estas figuras se focalizará en intentar regenerar toda la información contenida en éstas y hacer que los caracteres peor definidos sean visibles para el OCR. Al igual que los procesados realizados sobre las imágenes con modificaciones en el texto, se centrarán en aumentar el contraste entre los dos planos de la imagen.

De los procesados utilizados para modificar el contraste de la imagen, tanto la igualación de histograma como el ajuste de contraste, no han ofrecido mejoras en el porcentaje de éxito del OCR ninguno de ellos.

Continuamos con la aplicación de umbrales por diferentes sectores de la imagen y comprobar si obtenemos mejoras, de igual manera que ocurría en el caso de los archivos con deformaciones en el texto.

Pues bien, ni realizando la umbralización por áreas, que hasta ahora había resultado ventajoso en todos los casos, hemos conseguido ninguna mejora en nuestro sistema de detección de SPAM. Por lo tanto este hecho nos hace llegar a la conclusión de que la calidad de las imágenes que tenemos es tan pobre que el OCR no falla por las modificaciones que hayan sufrido sino por la calidad de éstas.



De cara al lector, los archivos están claros y se pueden leer sin problema, sin embargo para el OCR no lo son como muestra la figura próxima. La identificación del final y del comienzo de caracteres consecutivos pasa a ser una tarea de dificultad máxima. Destacar los casos más claros de lo comentado en las parejas de caracteres de la imagen *D* y *E* o *A* y *L*.

No obstante, debido a que el texto no había sufrido ningún cambio geométrico, podemos concluir el análisis práctico de los archivos con cambios en la estructura, dando unos pasos que nos pueden ser útiles a la hora de trabajar con estas imágenes:

1. Cambio de la imagen de color a escala de grises. Con este procesado se ha comprobado que obtenemos mejoras en nuestro sistema.

2. Procesados que generen una mayor diferencia entre los dos planos principales de la imagen. Con éstos, no hemos conseguido mejoras pero tal y como hemos comentado, la calidad de los archivos es la limitación que nos encontramos.

Aunque con las imágenes de este apartado las mejoras no han sido espectaculares, mostraremos el porcentaje de mejora obtenido con el cambio a escala de grises, que ha sido el único procesado que aumenta la tasa de acierto del OCR.

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Estructura1	43,03%	56,35%
Estructura2	38,51%	44,18%

Procesado de imágenes con transformaciones en el plano secundario:

En este subapartado del desarrollo experimental realizaremos distintos tipos de filtrados sobre las imágenes que hayan sufrido cambios en su plano secundario. Recordar que las modificaciones más importantes que pueden verse en el plano secundario son: colores, puntos, líneas, formas aleatorias y combinaciones.

- **Colores**:

Los archivos agrupados en este subapartado de modificaciones en el segundo plano se caracterizan en que el texto contenido en la imagen está diferenciado únicamente por el color del plano secundario. Este hecho hace que en ellos no podamos aplicar un cambio a escala de grises como habíamos estado haciendo hasta este momento ya que causaríamos que la información para diferenciar entre texto y fondo desapareciera.

No obstante, antes de comenzar con el procesado de este tipo de cambios, vamos a recordar las tasas de acierto que teníamos con estos archivos.

	Tasa acierto OCR Imagen Original
Colores1	32,31%
Colores2	71,88%
Colores3	0,00%
Colores4	5,37%
Colores5	71,72%
Colores6	0,00%

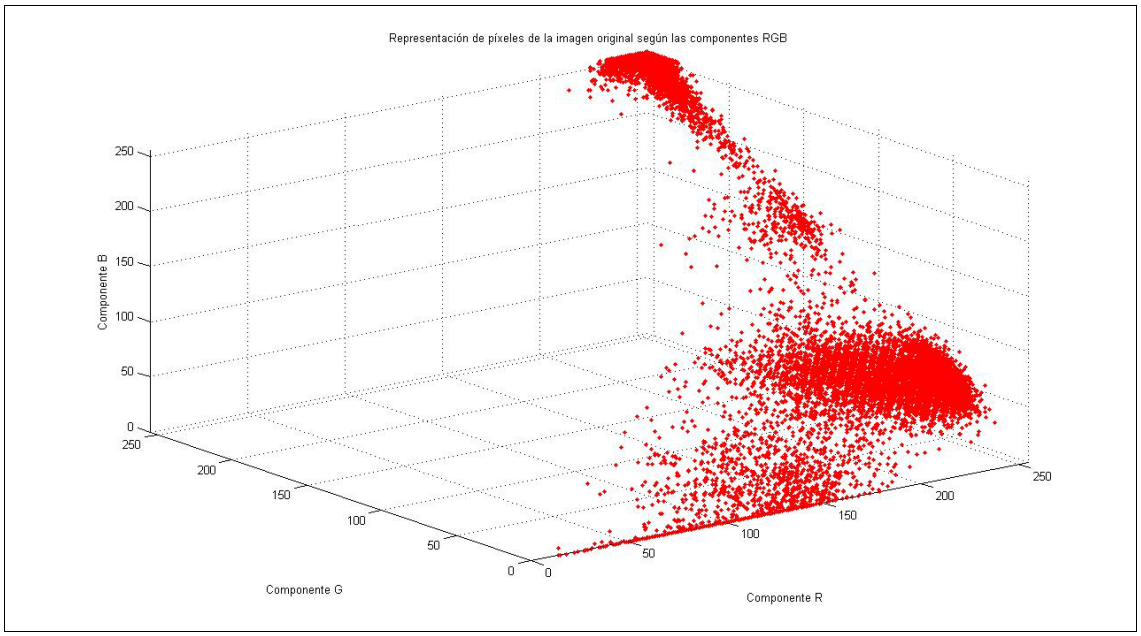
Para la explicación del procesado a desarrollar con las imágenes con modificaciones en el color, vamos a coger como ejemplo el archivo *Colores1*. Tener en cuenta que para el resto de imágenes el procesado se realizará de la misma manera.

En la imagen original *Colores1* apreciamos distintos tonos de color según se trate del fondo de la imagen, del texto contenido en ella o de la zona en color rojo

que dificulta el trabajo del OCR.

Con el procesado vamos a buscar que todos los píxeles de la imagen que tengan el mismo color aparezcan en una imagen independiente, llegando a una imagen por cada color que esté presente en ella. Para el ejemplo que nos ocupa, tendremos una imagen con los píxeles de color negro, otra para los de color blanco y la última para los de color rojo.

La siguiente figura es el resultado de representar, en el espacio RGB, la imagen que nos ocupa. Cada uno de los píxeles existentes en la imagen queda representado en la figura por un punto.



El siguiente paso a llevar a cabo en este procesado sería determinar los grupos o clusters de colores en los que queremos separar el archivo a tratar. Para ello, podemos utilizar el método *kmeans* de Matlab que se ocupará de darnos estos clusters.

Kmeans funciona de la siguiente manera: agrupa píxeles de forma que los que pertenezcan al mismo cluster, estén muy cercanos entre sí, mientras que se alejen lo máximo posible de píxeles que sean de otro cluster. Cada cluster queda definido por lo puntos que lo componen y por su centroide, que viene a ser el punto que hace que la suma de distancias entre él y los puntos de su cluster sea minimizada.

El problema principal del método *kmeans* es que hay que indicarle como parámetro de entrada el número de clusters finales que queremos tener. Con la imagen que estamos tratando es fácil de concluir que el número final de clusters o grupos que queremos tener por separado es de tres: uno para los píxeles de color negro, otro para los de rojo y otro para los de blanco. Sin embargo, determinar este número de grupos no es siempre tan sencillo, por lo que hay que idear un método que nos genere este número óptimo de grupos de forma analítica.

La búsqueda del número óptimo de clusters comenzará por introducir como parámetro de entrada al método *kmeans* un valor de dos clusters, que es el número

mínimo de grupos que puede poseer una imagen. Tras calcular los dos clusters, pasaremos a cambiar el número de grupos deseados a tres. Así sucesivamente hasta que se cumpla un criterio de parada que nos de el número óptimo de clusters.

El criterio de parada lo vamos a conseguir mediante el método de Matlab *silhouette*. Esta instrucción nos da una idea de cómo de cercano está cada punto de un cluster al resto de grupos. Por lo tanto, con hacer la media de esta silueta, tenemos un valor que podemos tomar para ver cuando llegamos al número de clusters óptimo.

Entonces, el procedimiento para calcular este número de clusters será:

1. Agrupar los píxeles de la imagen en N grupos mediante *kmeans*, inicialmente dos.

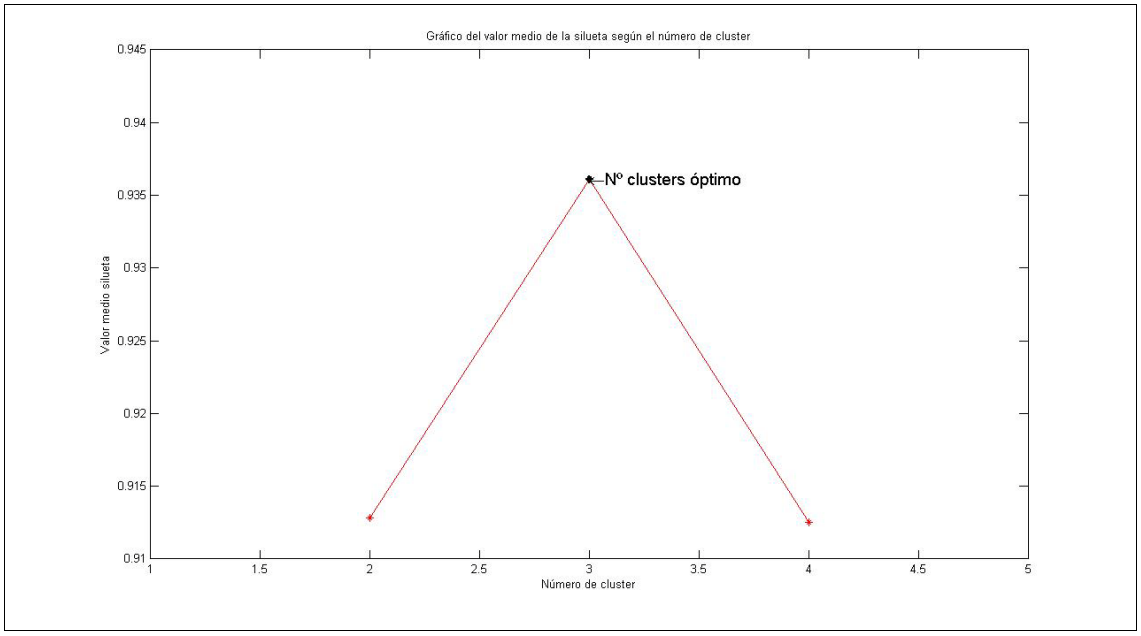
2. Calcular la media de la silueta con N grupos mediante *silhouette*, valor $MediaSilueta_N$.

3. Aumentar el número de grupos a N+1 y agrupar los píxeles en N+1 grupos mediante *kmeans*.

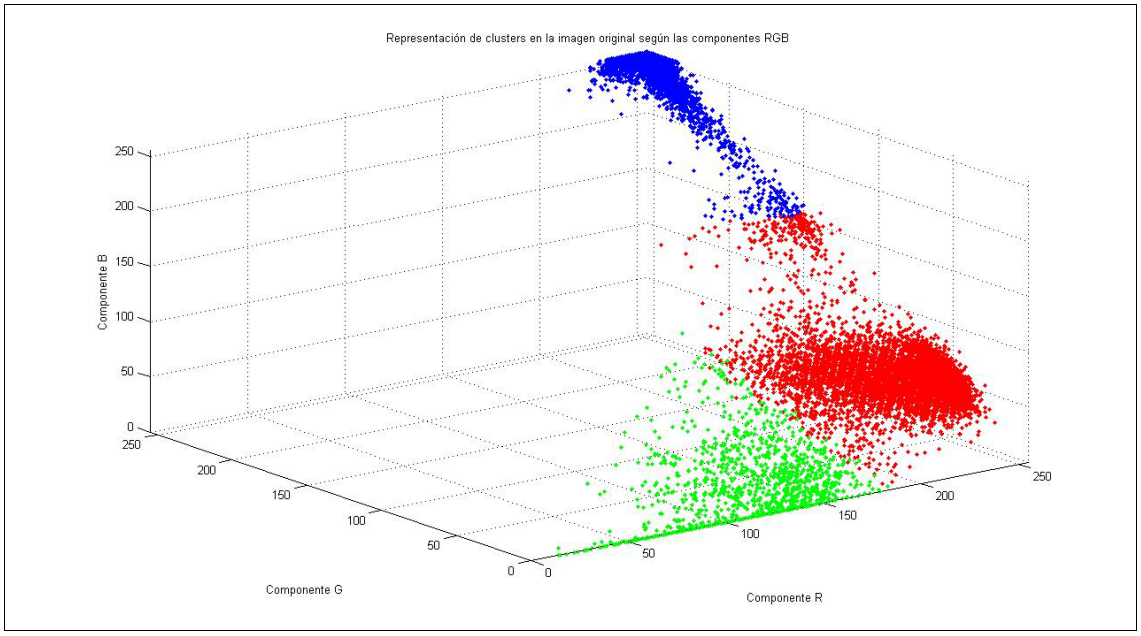
4. Calcular la media de la silueta con N+1 grupos mediante *silhouette*, valor $MediaSilueta_{N+1}$.

5. Si $MediaSilueta_N$ es mayor que $MediaSilueta_{N+1}$, entonces el nº óptimo de clusters es N, sino volvemos al punto 3 y continuamos el proceso.

Con la imagen que nos ocupa, se puede ver que el mejor número de grupos en que separar la imagen es tres ya que el mayor valor de la media de la silueta es cuando se fija en tres el número de clusters. Para el caso de dos y cuatro grupos, los valores de la media de la silueta son menores que en el caso de tres cluster, lo que indica que no son los valores óptimos para el número de grupos a calcular.

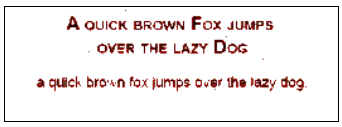


Una vez que hemos llegado a que el número de grupos en el que hay que separar nuestra imágenes es tres, procedemos a introducir este valor como parámetro de entrada al método *kmeans*, teniendo la siguiente figura como resultado:



En la figura previa se pueden diferenciar claramente los tres grupos creados con los píxeles de la imagen. Los distintos grupos han sido pintados con un color distinto para que se puedan apreciar de mejor manera cada uno de ellos.

En este punto del procesado, ya tenemos las subimágenes separadas por los distintos colores presentes en la imagen original.



Como se puede apreciar en el ejemplo anterior, ya hemos conseguido cada color por separado, que era nuestro objetivo de este procesado. Una vez que tenemos las subimágenes, podemos obtener la imagen que deseemos tan sólo con realizar sumas entre estas subimágenes.

Para cada una de las imágenes que tenemos que procesar, realizamos los mismos pasos que se han comentado hasta ahora. Las imágenes finales que vamos a introducir al OCR estarán compuestas por uno o más clusters que tendrán que ser elegidos por el usuario.

En la siguiente tabla se muestra el número de clusters en el que se ha dividido cada imagen, así como los clusters que forman parte de la imagen final introducida al OCR:

	Nº clusters finales	Imagen final compuesta por los clusters nº
Colores1	3	2
Colores2	2	2
Colores3	2	2
Colores4	5	2 y 4
Colores5	2	2
Colores6	3	1

Destacar que con todas las imágenes, el procesado ha sido capaz de detectar el número óptimo de clusters automáticamente. Sin embargo, para el archivo *Colores4* no ha podido detectar este número y ha tenido que ser fijado por el usuario el número de clusters.

	Tasa acierto OCR Imagen Original	Tasa acierto OCR Separación por colores
Colores1	32,31%	73,85%
Colores2	71,88%	71,88%
Colores3	0,00%	28,13%
Colores4	5,37%	97,31%
Colores5	71,72%	85,86%
Colores6	0,00%	76,47%

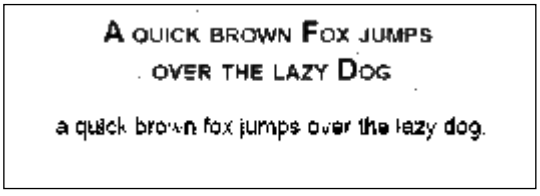
Como se puede apreciar en la tabla anexa, en todos los casos excepto en la imagen *Colores2*, la tasa de acierto del OCR tras la separación por colores ha crecido. Remarcar los casos de las imágenes *Colores4* y *Colores6*, en los que el porcentaje de acierto se ha visto una mejora por encima del 90% y 75% respectivamente.

En este punto, damos por finalizado el procesado para el archivo *Colores4* ya que la tasa de acierto alcanzada con él es prácticamente plena. Sin embargo, para el resto de casos vamos a seguir intentando que el porcentaje de éxito continúe creciendo.

Hemos conseguido, al separar los distintos colores presentes en las imágenes, quedarnos sólo con aquella parte de éstas que nos es más relevante. Entonces, ahora sí que podemos realizar un cambio a escala de grises de los archivos ya que no se producirá una pérdida de información debido a que sólo tenemos presente en ellas el texto que debe reconocer el OCR.

Con el cambio a blanco y negro anteriormente comentado, obtenemos que en los tres primeros casos (imagen *Colores1*, *Colores2* y *Colores3*) la tasa de acierto permanezca igual que antes de este cambio de bases. Sin embargo, para las imágenes *Colores5* y *Colores6* aumenta el porcentaje de acierto del OCR, llegando incluso en la última de ellas a no cometer ni un solo error. De esta forma, también se da por finalizado el análisis de la imagen *Colores6*.

Con las imágenes en blanco y negro, continuamos los procesados a aplicar en las imágenes. De igual forma que hemos procedido en los subapartados anteriores con otros tipos de modificaciones, vamos a aumentar el contraste entre el plano principal y secundario. Para ello vamos a aplicar umbrales, tanto en toda la imagen como por áreas y ver los resultados qué obtenemos.



En la única imagen en la que ha funcionado los procesados comentados en el párrafo anterior ha sido *Colores1*, en la que se ha aplicado un umbral por sectores con

valor de 79. En el resto de ellas, no hemos conseguido ninguna mejora en la tasa de acierto.

Además también se han probado los procesados para la igualación de histograma como el ajuste de contraste, no han ofrecido mejoras en el porcentaje de éxito del OCR ninguno de ellos.

Para finalizar con el tratamiento de este tipo de modificación sobre las imágenes, fijamos una serie de pasos a realizar para mejorar la tasa de acierto del OCR con ellos:

1. Separación de los colores presentes en la imagen original en subimágenes con un color cada una de ellas.

2. Transformación de la imagen de color a escala de grises.

3. Con la imagen en escala de grises, aplicar procesados que aumenten el contraste ente el texto del plano principal y el fondo de la imagen. El procesado que mejores resultados ha dado ha sido una umbralización por sectores.

Recalcar que con este tipo de modificaciones, el procesado que más beneficio nos otorga es la separación de colores en imágenes separadas.

La siguiente tabla muestra un resumen numérico de las tasas de acierto del OCR al comienzo y al final del procesado de estas imágenes.

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Colores1	32,31%	80,00%
Colores2	71,88%	71,88%
Colores3	0,00%	28,13%
Colores4	5,37%	97,31%
Colores5	71,72%	89,90%
Colores6	0,00%	100,00%

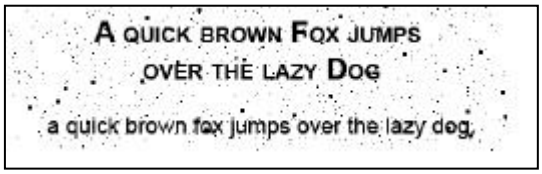
- **Puntos:**

Los archivos que poseemos con este tipo de modificación están caracterizados por la inclusión de puntos en el fondo de la imagen. De esta forma, se dificulta el correcto trabajo del OCR, causando que se produzcan tasas de acierto bajas, como las que se muestran a continuación.

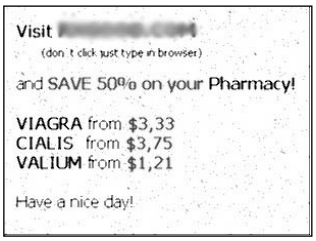
	Tasa acierto OCR Imagen Original
Puntos1	15,15%
Puntos2	36,04%

El primero de los procesados a realizar será el paso a escala de grises de las imágenes. Con éste, las tasas de acierto han sufrido un pequeño crecimiento.

	Tasa acierto OCR Imagen ByN
Puntos1	18,18%
Puntos2	42,36%



La modificación que ha sufrido este tipo de archivos podría asimilarse a la adicción de un ruido impulsivo en ellas. Los efectos negativos de este ruido se pueden eliminar teóricamente con un filtrado de mediana.



Para aplicar el filtro de mediana a estas imágenes utilizaremos una máscara cuadrada cuyo tamaño será especificado por el usuario. Para estas dos imágenes, el tamaño de máscara aplicado es de 2x2, haciendo que en la imagen *Puntos1* se produzca un desenfoque en ella, por lo que la tasa de acierto del OCR no aumenta. Sin embargo, en la imagen *Puntos2* no desaparece totalmente el ruido pero el porcentaje de éxito sí crece hasta el valor del 86,49%.

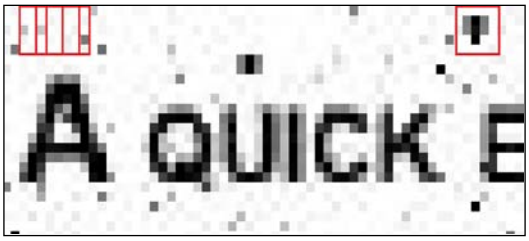
No obstante, aunque funcione el filtrado de mediana en una de las imágenes, vamos a buscar un procesado genérico que funcione sobre las dos imágenes y limpie éstas de los puntos de ruido.

El procesado buscado para eliminar los puntos de las imágenes consiste en lo siguiente: Se recorre la imagen con la máscara mostrada a continuación.

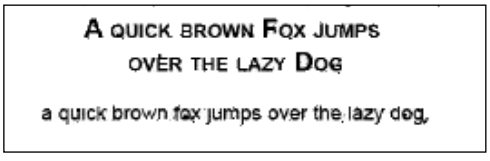
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1

Esta máscara se multiplica con cada bloque de 5x5 en la imagen, de forma que tenemos el perímetro del bloque con los valores de la imagen. Sobre este perímetro se aplica un umbral que es introducido por el usuario, de forma que si todos los valores del perímetro están por debajo del umbral, se pasa a blanco el bloque entero de 5x5 en la imagen.

En la imagen de ejemplo, al recorrer ésta con la máscara, cuando existe ruido, todo él está dentro del perímetro y es menor del umbral, por lo que se borra. Sin embargo, cuando llegamos a zonas con texto, el perímetro ya no es limpio y no se elimina nada.



Para este procesado, los umbrales utilizados para cada una de las imágenes han sido de 150 y 124 respectivamente. Se puede ver en la siguiente imagen, que todos los puntos que eran de ruido han desaparecido. Sin embargo, también han desaparecido los puntos de las "i"es ya que es imposible diferenciar cuales de los puntos son ruido y cuales pertenenecen al propio texto.



Como era de esperar, con esta limpieza en los archivos, las tasas de acierto del OCR han crecido para ambas imágenes, con aumentos en el porcentaje final del 60% y 45% respecto a la tasa previa a este procesado. En este momento los porcentajes se sitúan en el 77,27% para la imagen *Puntos1* y el 88,29% para la imagen *Puntos2*.

Una vez eliminado el ruido en estos archivos, podemos proceder de igual manera que en el resto de archivos con otras modificaciones, es decir, buscar procesados que hagan que el texto que nos interesa resalte más respecto al fondo.

Para buscar un mayor contraste entre el plano primario y el secundario se han probado los procesados para ajuste de contraste y la igualación de histograma, pero ninguno de ellos nos ha generado mejoras en la tasa de éxito del OCR.

De igual manera, hemos aplicado umbrales en ellas, tanto en toda la imagen como en distintos sectores, sin obtener aumentos en los porcentajes de acierto del OCR.

Por lo tanto, cuando nos encontremos con archivos que posean en el segundo plano puntos para dificultar el funcionamiento del OCR, los pasos a seguir para mejorar la tasa de acierto del OCR son:

1. Transformación de la imagen de color a escala de grises.

2. Eliminación de los puntos presentes en plano secundario de estos archivos.

3. Aplicar procesados que aumenten el contraste entre los dos planos de la imagen. Con los archivos del estudio no han sido necesarios pero en otros casos es probable que sí lo sean.

El procesado que más ventajas y que siempre hay que aplicar en este tipo de modificaciones es aquel que elimina los puntos en el plano secundario.

En resumen, las tasas de acierto del OCR antes y después de los procesados de las imágenes son las mostradas en la tabla siguiente:

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Puntos1	15,15%	77,27%
Puntos2	36,04%	88,29%

- **Líneas:**

Este tipo de modificación se diferencia de la anterior en que ésta inserta líneas de colores en el fondo de las imágenes. No obstante, las tasas de acierto del OCR de los archivos originales con las líneas presentes son bastante altas, estando en ambos casos por encima del 70%.

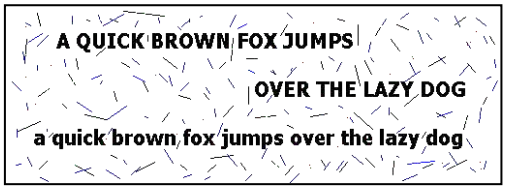
	Tasa acierto OCR Imagen Original
Lineas1	71,21%
Lineas2	72,73%

Los archivos que nos vamos a encontrar tendrán un aspecto similar al de la imagen anexa. Por lo que nuestro primer objetivo con ellas será eliminar las líneas que se encuentran en el fondo de la imagen.

El primer procesado que vamos a realizar sobre estas imágenes será el mismo que hemos aplicado en los archivos con modificaciones de color. Recordar que este procesado consistía en que cada color presente en la imagen de partida se representaba en una subimagen individual, para posteriormente quedarnos con las subimágenes que nos interesasen en cada caso.

Los parámetros generados por el procesado anterior son los siguientes:

	Nº clusters finales	Imagen final compuesta por los clusters nº
Lineas1	7	1 y 3
Lineas2	7	2 y 7

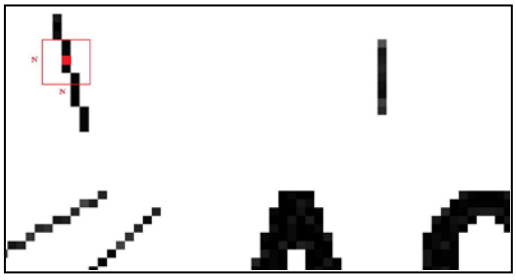


Pues bien, con este procesado, como se puede ver en la imagen del lateral, se ha reducido el número de líneas existentes en el segundo plano. Sin embargo, no se han eliminado todas estas formas rectas ya que existen algunas con el mismo color que el texto, por lo que agrupando por colores no pueden ser discriminadas en su totalidad.

El procesado previo sería muy efectivo en el caso de que todas las líneas del segundo plano fueran de un color distinto que el del texto que nos interesa. Si no es así, se puede considerar este procesado como el primero a aplicar pero es necesario continuar con otros. La imagen a procesar en los siguientes filtrados será la obtenida con el agrupamiento por colores.

Comprobado que con el agrupamiento por colores sobre la imagen original no conseguimos limpiar totalmente la imagen de ruido, pasamos a buscar filtrados sobre la imagen y escala de grises.

Entonces, el siguiente paso a realizar será el cambio de las imágenes agrupadas por colores a escala de grises. Este es un ejemplo de cómo quedarían los archivos con el cambio de bases.

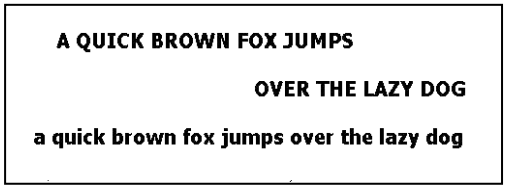


En este momento, pasamos al análisis del procesado que eliminará las líneas del fondo de las imágenes. El procesado consistirá en recorrer cada punto de la imagen y contabilizar el número de píxeles que no son blancos dentro de un recinto delimitado por una máscara de tamaño NxN. Si el número de

píxeles dentro de la máscara es menor que un umbral fijado por el usuario, se considera que ese píxel central pertenece a una línea y se elimina, pasándolo a color blanco. Si no es así, el píxel permanece sin cambios y el proceso continúa con el resto de puntos de la imagen.

Para el procesado previo, el usuario debe introducir dos parámetros: uno de ellos es el tamaño de la máscara que fijará el perímetro que influirá en cada píxel y el otro, el umbral para considerar que un punto forma parte de una línea o no. Los valores de estos parámetros son: para el archivo *Lineas1*, tamaño de máscara 5 y umbral 9 y el archivo *Lineas2*, tamaño de máscara 11 y umbral 10.

Observando los dos archivos que tenemos para el estudio de esta modificación, podemos ver que, en el segundo de ellos, el grosor del texto y de las líneas a eliminar es muy similar. Este hecho hace que el tamaño de la máscara tenga que ser mayor ya que necesitamos conocer una zona más grande alrededor del píxel para determinar si pertenece a una línea o no. Sin embargo, en la primera de ellas, como difiere de gran manera los grosores del texto y de las líneas, este tamaño de máscara es menor.



Con este procesado hemos conseguido que sean eliminadas de forma completa las líneas del plano secundario que impedían el correcto funcionamiento del OCR como muestra la imagen anexa.

Las tasas de acierto, tras eliminar la perturbación que nos ocupaba en este apartado, han crecido de manera considerable. En la imagen *Lineas1*, el acierto del OCR es pleno y en la *Lineas2*, la tasa de acierto se ha situado en un valor del 88,81%.

El análisis del archivo *Lineas1* se da por finalizado ya que el OCR reconoce correctamente todos los caracteres presente en ella. Para la imagen *Lineas2* continuamos con alguno de los procesados para aumentar el contrate entre los dos plano de la imagen y observar sus resultados.

Los procesados probados con el archivo *Lineas2* han sido igualación de histograma, umbral y ajuste del contraste, y ninguno de ello nos ha generado una mayor tasa de acierto del OCR, por lo que también damos por finalizado el análisis de esta imagen.

De igual manera que hemos realizado con el resto de modificaciones estudiadas hasta este punto, fijaremos un proceso a aplicar cada vez que tengamos que trabajar con imágenes en cuyo fondo se sitúan líneas de colores.

1. Separación de los colores presentes en la imagen original en subimágenes con un color distinto cada una de ellas. Este procesado será óptimo siempre que las líneas del fondo sean de diferente color que el del texto presente. Si no, debemos continuar con los siguientes procesados.

2. Transformación de la imagen resultante tras la separación por colores a una imagen en escala de grises.

3. Procesados para eliminar las líneas restantes en el fondo de la imagen.

Tener en cuenta que si tras la separación por colores hemos conseguido que desaparecieran todas las líneas del segundo plano, no serían necesarios el resto de procesados.

Como resumen final del estudio de esta modificación, facilitamos la siguiente tabla con el porcentaje de éxito del OCR antes y después de los procesados:

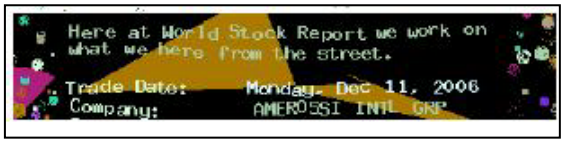
	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Lineas1	71,21%	100,00%
Lineas2	72,73%	88,81%

- **Formas aleatorias:**

Los archivos que vamos a procesar en este apartado se caracterizan en que, en el segundo plano de ellos, existen formas muy diversas que no se pueden considerar ni como puntos ni como líneas. Recalcar que las tasas de acierto obtenidas con los dos archivos originales son muy distintas entre ellas, siendo en una de ellas casi plena mientras que en la otra, el acierto es muy bajo.

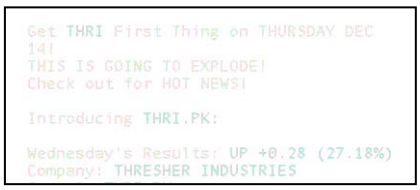
	Tasa acierto OCR Imagen Original
FormaAleatoria1	98,03%
FormaAleatoria2	16,35%

Con un estudio previo de las imágenes, podemos ver que en ambos casos, las letras del texto son de distinto color que los colores que están presentes en el fondo. Entonces, el primero de los procesados sería el mismo que el utilizado en el caso de imágenes con perturbaciones en el color, buscando separar cada uno de los colores de la imagen en otras subimágenes.



El número de clusters para cada uno de los archivos, así como los clusters elegidos para la imagen final, se muestran en el cuadro posterior:

	Nº clusters finales	Imagen final compuesta por los clusters nº
FormaAleatoria1	2	2
FormaAleatoria2	7	1, 2, 6 y 7



Para la imagen *FormaAleatoria1*, este proceso funciona perfectamente y separa de forma correcta en una subimagen el texto y en otra el fondo. Sin embargo, debido a que la tasa de acierto del OCR con la imagen de partida era tan alta, no conseguimos aumentarla y se siguen produciendo los mismos fallos que antes de la separación por colores. Entonces, el análisis de este archivo lo podemos dar por finalizado ya que no vamos a conseguir ninguna mejora con el resto de procesados.

En cambio, para la imagen *FormaAleatoria2* este procesado no es tan óptimo. La tasa de acierto del OCR ha mejorado y se sitúa en un valor del 25,96%. Sin embargo, podemos ver que hay parte del texto que se ha separado correctamente pero parte que no. Aplicando un zoom en la imagen original podemos analizar cuál es el problema en esta imagen.

Podemos ver como en la imagen original existen zonas en las que es imposible diferenciar qué parte pertenece al texto y qué parte al fondo de la imagen, ya que los colores se han mezclado. Si esta misma imagen tuviera los límites entre texto y fondo bien delimitados y no con tan mala calidad, este procesado para separar por colores sería el más indicado.

Por lo tanto, debido al problema comentado anteriormente, el porcentaje de éxito que se obtendrá con el archivo *FormaAleatoria2* no será muy alto ya que con la separación de colores se ha producido una pérdida de información.

Como continuación del análisis, seguiremos con un cambio de base a escala de grises sobre la imagen resultante tras la separación de colores. Con esta transformación de bases, ya tendremos la imagen en blanco y negro, por lo que podremos aplicar filtrados para aumentar el contraste entre plano e intentar que la tasa de acierto de nuestro sistema crezca.

Con la imagen *FormaAleatoria2* en escala de grises han sido probados los procesados para igualar el histograma, ajustar el contraste y un umbral. Tanto al aplicar el umbral con valor 251 como al ajustar el histograma de la imagen se ha conseguido mejoras en los resultados pero el mayor aumento del porcentaje de éxito se ha alcanzado con la igualación del histograma, situándose la tasa final en un 45,19%.

De cara a fijar unos pasos a utilizar siempre que estemos delante de imágenes con formas aleatorias en el fondo, se presentan los siguientes puntos:

1. Separación de los colores presentes en la imagen original en subimágenes con un color distinto cada una de ellas. Funcionará este procesado de forma correcta cuando el texto tenga diferente color que los presentes en el fondo y cuando exista un límite claro entre texto y fondo. Con estos requisitos cumplidos, este procesado es más que suficiente y no habría que aplicar los siguientes.

2. Transformación de la imagen resultante tras la separación por colores a una imagen en escala de grises.

3. Procesados para aumentar el contraste entre el primer y segundo plano. En este caso el que mejores resultados ha ofrecido ha sido la igualación del histograma.

Finalizado el procesado de estos archivos y fijados los pasos a seguir cuando nos encontremos con uno de ellos, mostramos los resultados obtenidos con nuestro análisis:

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
FormaAleatoria1	98,03%	98,03%
FormaAleatoria2	16,35%	45,19%

- **Combinaciones:**

En este caso, las imágenes poseen en su fondo una combinación de las modificaciones estudiadas en los apartados anteriores, líneas, formas aleatorias y puntos.

Las tasas de acierto que alcanza el OCR con las imágenes originales son las mostradas en la tabla siguiente:

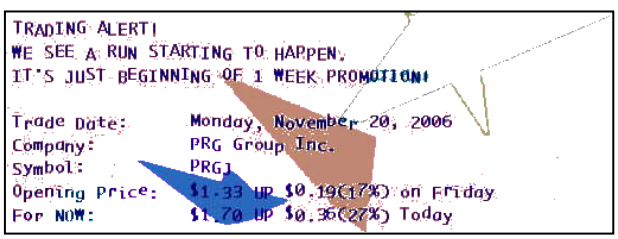
	Tasa acierto OCR Imagen Original
Combinación1	67,72%
Combinación2	90,51%
Combinación3	75,66%
Combinación4	0,00%

De igual forma que hemos realizado en los apartados de líneas y formas aleatorias, el primero de los procesados será para separar cada uno de los colores presentes en la imagen en subimágenes.

La separación por colores de cada una de las imágenes se ha visto caracterizada por los siguientes valores del número de clusters, así como los grupos que forman parte de la imagen final.

	Nº clusters finales	Imagen final compuesta por los clusters nº
Combinación1	6	4, 5 y 6
Combinación2	6	2, 4 y 6
Combinación3	6	2, 5 y 6
Combinación4	9	4, 6 y 7

La separación por colores en los casos *Combinación1*, *Combinación2* y *Combinación3* ha funcionado correctamente y se ha conseguido separar el texto.



Sin embargo, con la imagen *Combinación4* no se han obtenido los mismos resultados que con el resto de imágenes, ya que posee el mismo problema que tenía el archivo *FormaAleatoria2*, es decir, no existe la

suficiente diferencia entre colores colindantes como para que no se produzca pérdida de información al quedarnos con un color o con otro. No obstante, prácticamente se ha recuperado todo el texto aunque existen zonas del fondo que no se han podido eliminar.

Debido a que estas imágenes habían sido modificadas mediante diversas formas, con la separación por colores hemos eliminado una de ellas pero continúan existiendo otras modificaciones.

A continuación vamos a realizar sobre las imágenes separadas por colores un cambio de escala de grises, para posteriormente sobre los archivos en blanco y negro aplicar el resto de procesados para deshacernos de las perturbaciones aún no eliminadas.

Debido a que con las imágenes que nos encontramos en este apartado han sufrido una combinación de alteraciones, los procesados a aplicar sobre ellas también serán combinación de los estudiados hasta ahora. En la siguiente tabla se hace un resumen con los procesados aplicados en cada imagen para eliminar las modificaciones que hubieran sufrido, así como los parámetros usados en los procesados.

	Imagen original	Procesados aplicados	Parámetros
Combinación1	ByN	Ninguno	
Combinación2	ByN	Eliminación líneas	Tamaño de máscara 5 Umbral 9
Combinación3	ByN	Eliminación ruido Eliminación líneas	Umbral de ruido 200 Tamaño de máscara 19 Umbral 22
Combinación4	ByN	Ninguno	

Como se puede ver, en dos de las imágenes no se ha aplicado ningún procesado ya que éstas, en su segundo plano, no poseían ni líneas ni ruido que hubiese que eliminar, como en cambio sí ocurría en las otras dos.

Las tasas de acierto del OCR para las imágenes en las que se ha eliminado las perturbaciones en el segundo plano han aumentado hasta los valores de 97,38% para el archivo *Combinación2* y de 84,65 para el archivo *Combinación3*.

Una vez que hemos conseguido que los archivos estén limpios de modificaciones en el segundo plano de ellas, pasamos a intentar resaltar más la diferencia entre el texto presente y el fondo.

Para hacer más visible la diferencia entre la información del primer plano y el segundo plano, hemos probado la igualación del histograma y el ajuste de contraste pero en ninguna de ellas se ha generado una mejora en nuestro sistema. Además también se han aplicado los procesados mediante umbrales, tanto por sectores de la imagen como por toda ella y no se han conseguido mejores resultados, exceptuando con el archivo *Combinación4*. Con este archivo, la tasa de acierto ha crecido hasta el 86,14% con los valores de umbral de 68 para el umbral por áreas y de 115 para el umbral en toda la imagen.

Finalizando el estudio de estos archivos, damos unas reglas a aplicar de cara al procesado de este tipo de modificaciones en las imágenes:

1. Separación de los colores presentes en la imagen original en subimágenes con un color distinto cada una de ellas. Si todas las perturbaciones existentes en el segundo plano son de un color diferente que el texto, este procesado sería el único a realizar ya que tendríamos el texto en una imagen independiente y separado de los elementos del fondo.

2. Transformación de la imagen resultante tras la separación por colores a una imagen en escala de grises.

3. Procesados para eliminar las perturbaciones aún presentes en la imagen en blanco y negro. Como estamos en el apartado de combinaciones, existen diferentes cambios, por lo que es necesario aplicar una combinación de procesados. Éstos pueden ser para eliminar el ruido, las líneas, los puntos…

4. Procesados para aumentar la diferencia entre los dos planos. No son siempre necesarios pero en algunos casos hacen que los resultados mejoren de forma muy importante.

El porcentaje de acierto de nuestro sistema tras todos los procesados llevados a cabo se muestra en la siguiente tabla. En ella se puede ver la mejora que hemos obtenido con el análisis de estas imágenes.

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Combinación1	67,72%	73,18%
Combinación2	90,51%	97,38%
Combinación3	75,66%	84,65%
Combinación4	0,00%	86,14%

Procesado de imágenes con transformaciones en el formato:

Este apartado del análisis experimental se centrará en buscar mejoras en las prestaciones en nuestro sistema de detección de SPAM para aquellas imágenes que han sufrido cambios en el formato. Recordar que estos cambios en el formato podían ser deformación en los archivos gráficos o corte del texto que aparece en ellos.

- **Deformación:**

La principal característica de estos archivos es que el tamaño de ellos se iba reduciendo de forma progresiva, a la vez que la tasa de acierto del OCR iba disminuyendo de igual forma en los que la complejidad era mayor. En aquellos que eran más sencillos, no se podía apreciar de forma tan clara la relación entre tamaño y éxito del OCR.

El número de imágenes de las que disponemos con este tipo de modificación es de seis pero sin embargo, el desarrollo experimental con filtrados únicamente lo vamos

a desarrollar para cinco ya que con el archivo *Deformación5* se alcaza un éxito total desde el principio del proyecto y no tiene ningún sentido aplicar nuevos filtrados sobre él.

	Tasa acierto OCR Imagen Original
Deformación1	95,58%
Deformación2	85,21%
Deformación3	46,58%
Deformación4	93,33%
Deformación5	100,00%
Deformación6	91,11%

El primer procesado que vamos a realizar sobre este tipo de imágenes será un cambio de la imagen a color a una imagen en escala de grises ya que el texto presente en ellas no es diferenciable de la perturbación por el color.

Con las tres primeras imágenes podemos ver que la tasa de acierto más alta se alcanza con aquella que posee un mayor tamaño (*Deformación1*), lo que nos puede indicar que si conseguimos un tamaño más grande en todas ellas, el porcentaje de acierto será similar al de la imagen *Deformación1*. De igual forma vamos a proceder con las otras dos imágenes, haciendo que la imagen *Deformación6* tenga unas dimensiones parecidas a al archivo *Deformación4*.

Para el cambio de tamaño en las imágenes nos basamos en el método *imresize* de Matlab que nos genera una expansión o compresión de la imagen original.

Según habíamos planteado la hipótesis, un mayor tamaño de la imagen nos iba a generar una mayor tasa de acierto, pues bien, no solo no se mejora el porcentaje de acierto del OCR sino que se empeora. Este hecho se debe a que, en el proceso de expansión, todos aquellos píxeles que han sido creados nuevos deben tomar un valor que el programa ha de generar. Este valor se calcula de forma estadística con los píxeles cercanos a él y como vemos, se introduce nueva información en la imagen que hace que ésta se degrade, de igual forma que lo hace el porcentaje de éxito.

Con el procesado anterior hemos intentado corregir las perturbaciones que estos archivos habían sufrido en el formato. Sin embargo, hemos visto que no hemos conseguido buenos resultados, por lo que los únicos procesados que pueden hacer que aumenten la tasa de nuestro sistema son aquellos que realcen la diferencia entre el texto y el segundo plano.

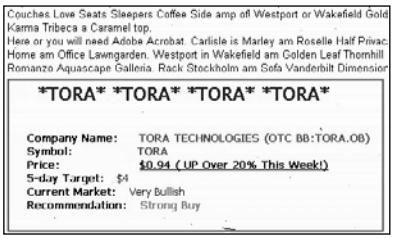
Tanto la igualación de histograma como el ajuste de contraste no nos han generado ninguna mejora en las prestaciones de nuestro sistema, por lo tanto pasamos a aplicar un umbral en las imágenes.

	Umbrales aplicados
Deformación1	155
Deformación2	171, 202, 158 y 213
Deformación3	199, 226, 199 y 207
Deformación4	206
Deformación6	194

Los umbrales que hemos aplicado a las distintas imágenes han sido introducidos por el usuario, teniendo los valores que se muestran en la tabla anexa. En aquellas imágenes que parecen varios valores de umbral se debe a que se han aplicado sucesivos umbrales por distintas áreas.

Con los procesados mediante umbral, tanto en toda la imagen como por áreas, hemos conseguido mejorar la tasa de acierto del OCR en los archivos *Deformación2*, *Deformación3* y *Deformación6*, en los otros dos casos no se ha visto alterada la tasa de acierto. Estos son los porcentajes conseguidos con dichos filtrados:

	Tasa acierto OCR Imagen Original
Deformación1	95,58%
Deformación2	89,95%
Deformación3	61,37%
Deformación4	93,33%
Deformación6	100,00%



En este punto podríamos dar por finalizados los procesados a realizar sobre estas imágenes, sin embargo, observando las tres primeras imágenes, podemos ver que hay parte del texto que está subrayado. Como estudiamos en el apartado de Texto Deformado, el subrayado hace que la tasa de éxito baje, por lo que vamos a proceder a eliminar este subrayado.

Recordar que el procesado necesitaba un umbral utilizado para borrar los subrayados. Los valores del umbral son: *Deformación1* umbral de 127, *Deformación2* umbral de 205 y *Deformación3* umbral de 191.

Ahora ya sin estar presente el subrayado en los archivos, la tasa de acierto del OCR ha aumentado en todos los casos, exceptuando la imagen *Deformación1* que ha permanecido constante.

De cara a dar por finalizado el análisis de estas imágenes con perturbaciones en el formato, vamos a proceder a fijar los procesados a realizar cuando estemos ante una imagen con este tipo de perturbación:

1. Cambio de la imagen original a una imagen en escala de grises.

2. Procesados que hagan que el contraste entre el primero y segundo plano aumente. De este tipo de procesados, el que mejor resultado ha dado es la umbralización.

3. Observando los archivos, podemos ver aplicar distintos procesados que se adecuen a las modificaciones que sufran las imágenes. Para las tres primeras, eliminación del subrayado del texto.

Las tasas de acierto finales tras los procesados previos para cada una de las imágenes son las mostradas en la siguiente tabla:

	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Deformación1	95,58%	95,58%
Deformación2	85,21%	90,73%
Deformación3	46,58%	66,23%
Deformación4	93,33%	93,33%
Deformación5	100,00%	
Deformación6	91,11%	100,00%

- **Corte:**

En este tipo de transformación, el texto de las imágenes se encuentra dividido en distintas partes, lo que causa que el OCR funcione de la peor forma posible y genere tasas de acierto nulas en las dos imágenes.

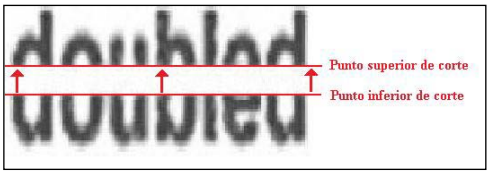
	Tasa acierto OCR Imagen Original
Corte1	0,00%
Corte2	0,00%

En este tipo de modificaciones no es relevante la información del color, por lo que se puede despreciar y fijar el primero de los procesados a realizar la transformación a escala de grises de las imágenes.

Como era de esperar, con el cambio de los archivos a blanco y negro, los porcentajes de éxitos del OCR no se han visto alterados y continúan siendo nulos.

El procesado en el que nos vamos a centrar a continuación es el más importante que vamos a tener que aplicar sobre estos archivos, ya que es aquel que consigue eliminar los cortes presentes en el texto.

Para deshacernos de esta perturbación la forma de actuar será la siguiente: primero el usuario, señalará el punto superior de corte del texto y posteriormente, el punto inferior. Una vez que tenemos los puntos de corte marcados, seleccionamos la parte de la imagen desde el punto inferior hasta el límite inferior de ésta. Únicamente nos falta desplazar esta parte seleccionada de la imagen hasta el límite superior, quedando de esta forma el corte eliminado.



Con las imágenes que tenemos para esta perturbación, han quedado los siguientes puntos superiores o inferiores definidos por el usuario: *Corte1* punto superior 22 y punto inferior 40, *Corte2* puntos superiores 33 y 58 y puntos inferiores 36 y 61. Para la segunda de las imágenes han sido necesarias dos iteraciones ya que el texto estaba segmentado en tres partes, de ahí que haya dos parejas de puntos superiores e inferiores.

En este momento, todo el texto contenido en los archivos es claramente legible, lo que hace que las tasas de acierto del OCR hayan aumentado de forma espectacular. Los valores para estas tasas son del 85,71% para el archivo *Corte1* y del 95% para el archivo *Corte2*, teniendo en cuenta que partíamos de una tasa nula, significa un aumento muy considerable.

Una vez que hemos eliminado los cortes en el texto, podemos pasar a probar otra serie de procesados que hagan que crezca aún más la tasa de acierto del OCR.

Mediante la umbralización, hemos conseguido que el porcentaje de éxito para la imagen *Corte1* sea total, pero para la imagen *Corte2* permanece constante. Los valores de umbral que hemos aplicado son 165 y 214, respectivamente para cada archivo.

Al igual que hemos realizado con el resto de apartados, vamos a fijar los pasos a seguir cuando los archivos a tratar tengan cortes en el texto:

1. Transformación de la imagen original a una imagen en escala de grises.

2. Eliminar los cortes del texto presentes en la imagen.

3. Procesados para aumentar la diferencia entre el texto y el segundo plano de la imagen. En este caso hemos aplicado un umbral a toda la imagen.

Destacar que el procesado que hace que aumente de forma espectacular la tasa de acierto del OCR es aquel que consigue eliminar los cortes presente en el texto de las imáganes.

Para finalizar este análisis, mostramos los porcentajes de éxito de nuestro sistema, antes y después de los procesados llevados a cabo. Hemos conseguido una mejora por encima del 95% en ambas imágenes con el procesado digital.

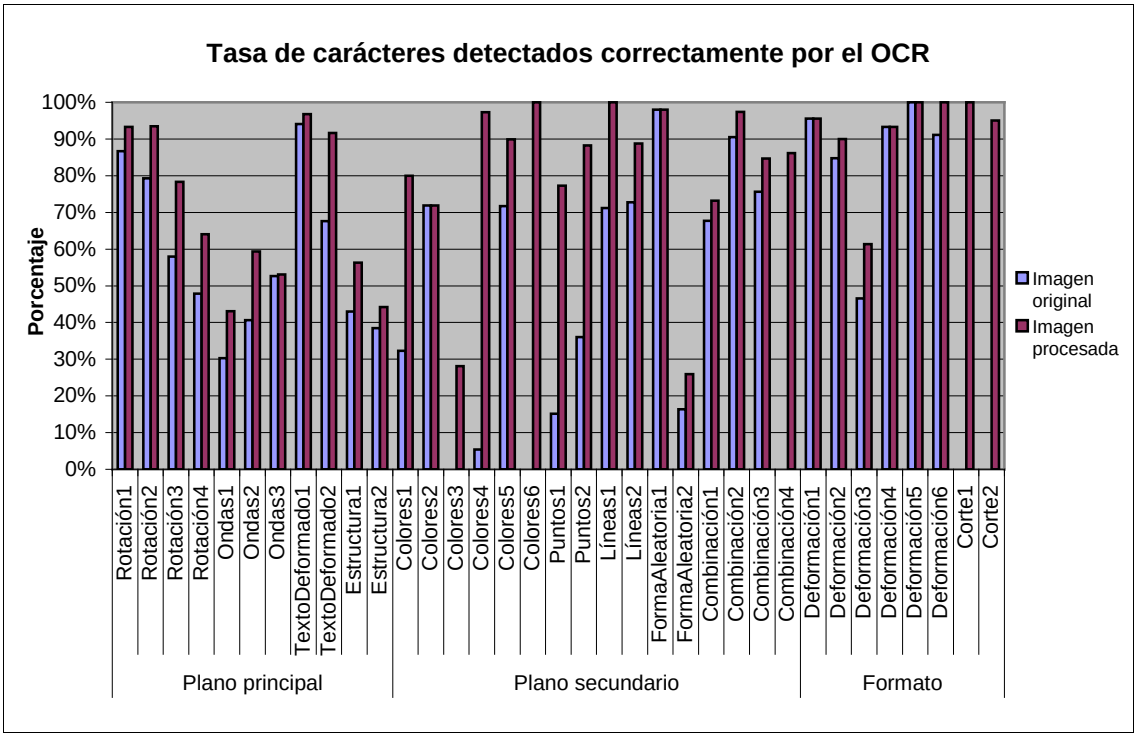
	Tasa acierto OCR Imagen Original	Tasa acierto OCR máxima alcanzada
Corte1	0,00%	100,00%
Corte2	0,00%	95,00%

5.2 Resultados finales tras el procesado

Una vez alcanzado este punto del proyecto, vamos a presentar los resultados alcanzados tras el correspondiente procesado digital de las imágenes que hemos llevado a cabo justo antes, en el apartado anterior de Estudio experimental.

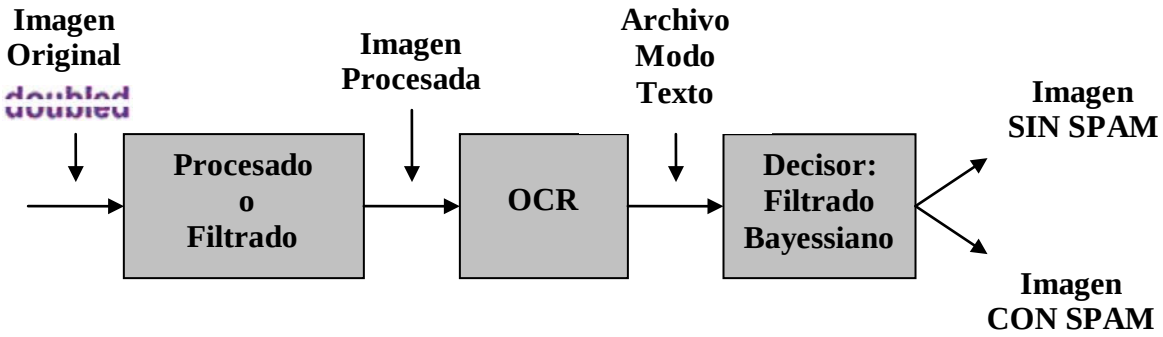
En este subapartado únicamente mostraremos los resultados obtenidos, dejando la presentación de las conclusiones para el siguiente apartado de nuestro trabajo.

La siguiente figura muestra las tasas de acierto del OCR que teníamos con las imágenes originales del comienzo del proyecto y los porcentajes más altos que hemos alcanzado después del procesamiento digital de éstas.

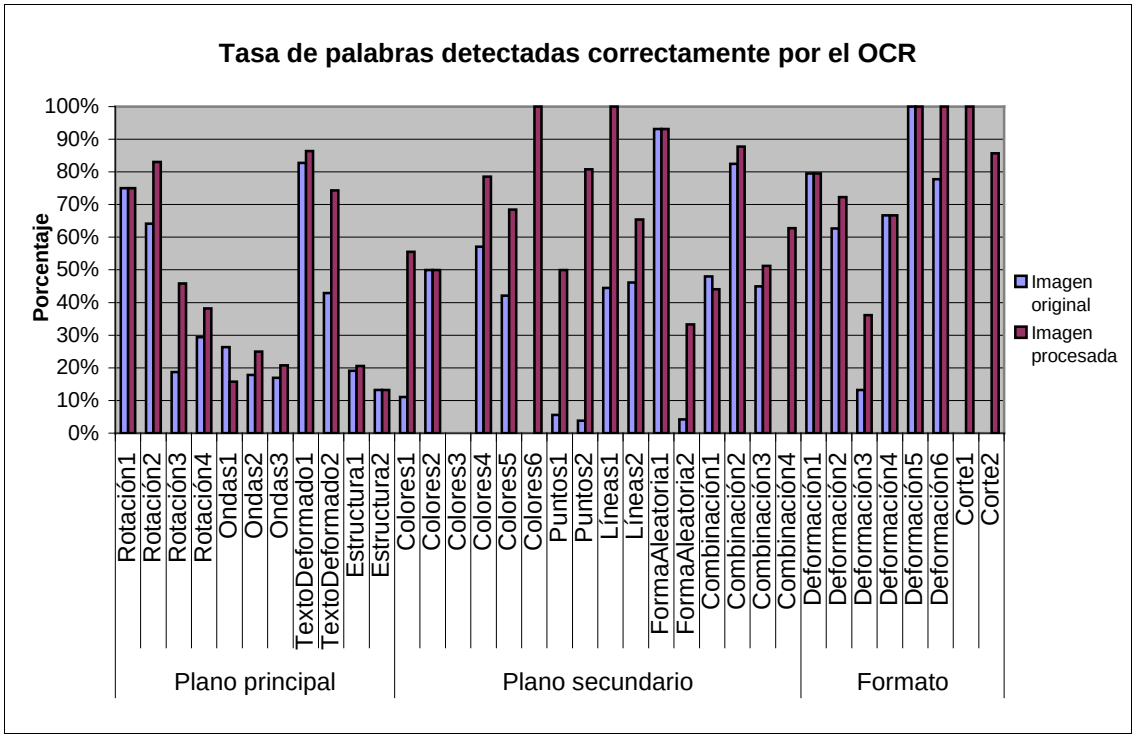


Como era de esperar, en la mayoría de los casos hemos conseguido una mejora del porcentaje de caracteres detectados correctamente por el OCR respecto al valor de partida que poseíamos. Si no fuera así, todo el trabajo realizado en el procesado de estas imágenes no hubiera tenido ningún sentido.

Por otra parte, volviendo al esquema de partida del sistema de detección de SPAM gráfico con OCR, podemos ver que el siguiente paso después del OCR es el decisor. A este decisor le introducimos el archivo modo texto generado por el OCR y es capaz de diferenciar entre palabras que sean SPAM y aquellas que no lo sean.



Por lo tanto, como hemos explicado anteriormente, también nos puede ser de utilidad el porcentaje de palabras acertadas correctamente por el OCR, tanto antes como después de los procesados. El siguiente gráfico mostrará los resultados obtenidos en cuanto a porcentaje de palabras acertadas se refiere.



Como hemos comentado previamente, todas las conclusiones a las que hemos llegado gracias a estos resultados las presentaremos en el siguiente apartado.

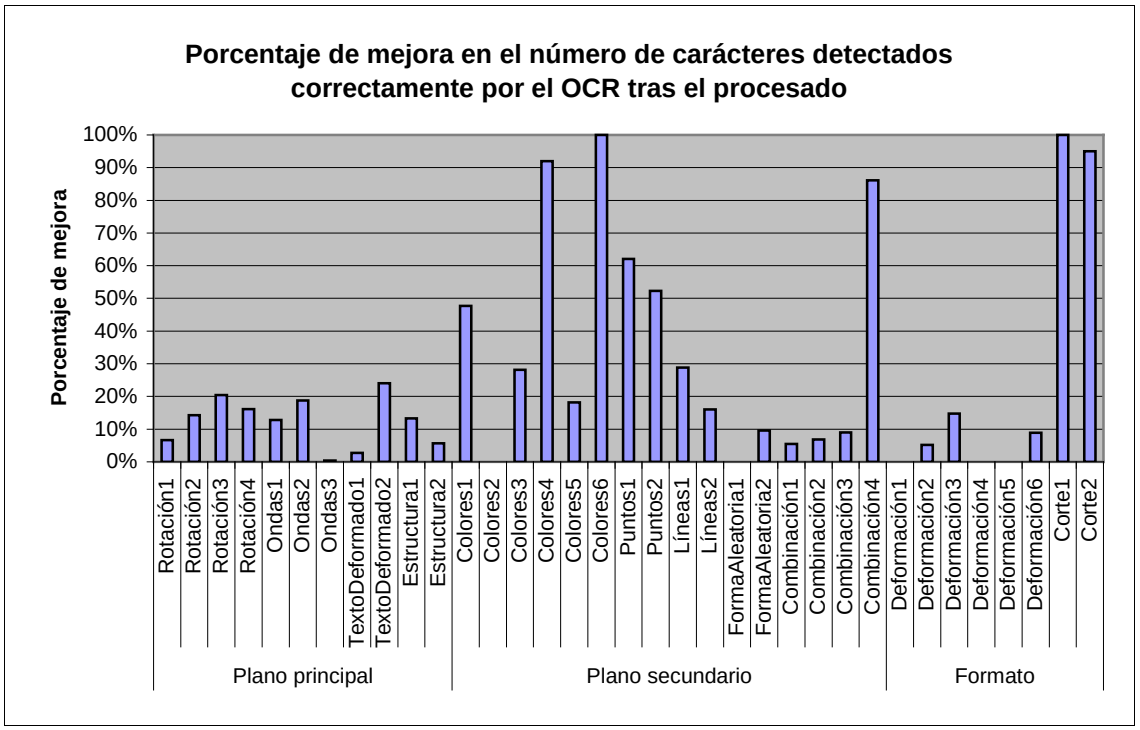
En la siguiente página mostramos la tabla resumen con los valores más significativos de porcentajes que hemos obtenido durante el desarrollo de los distintos procesados a lo largo del apartado de estudio experimental.

Transformaciones	Imagen	Porcentaje de carácteres detectados correctamente OCR imagen original	Porcentaje máximo alcanzado de carácteres detectados correctamente OCR tras procesamiento	Mejora del porcentaje de carácteres detectados correctamente OCR tras procesamiento	Porcentaje de palabras detectadas correctamente OCR imagen original	Porcentaje de palabras detectadas correctamente OCR tras procesamiento	Mejora del porcentaje de palabras detectadas correctamente OCR tras procesamiento
Plano principal	Rotación1	86,67%	93,33%	6,66%	75,00%	75,00%	0,00%
	Rotación2	79,27%	93,50%	14,23%	64,15%	83,02%	18,87%
	Rotación3	57,92%	78,33%	20,41%	18,75%	45,83%	27,08%
	Rotación4	47,90%	64,07%	16,17%	29,41%	38,24%	8,83%
	Ondas1	30,28%	43,12%	12,84%	15,79%	10,53%	-5,26%
	Ondas2	40,63%	59,38%	18,75%	17,86%	25,00%	7,14%
	Ondas3	52,65%	53,06%	0,41%	16,98%	20,75%	3,77%
	TextoDeformado1	94,06%	96,78%	2,72%	82,72%	86,42%	3,70%
	TextoDeformado1	94,06%	96,78%	2,72%	82,72%	86,42%	3,70%
	Estructura1	43,03%	56,35%	13,32%	19,12%	20,59%	1,47%
	Estructura2	38,51%	44,18%	5,67%	13,24%	13,24%	0,00%
Plano secundario	Colores1	32,31%	80,00%	47,69%	11,11%	55,56%	44,45%
	Colores2	71,88%	71,88%	0,00%	50,00%	50,00%	0,00%
	Colores3	0,00%	28,13%	28,13%	0,00%	0,00%	0,00%
	Colores4	5,37%	97,31%	91,94%	57,14%	78,57%	21,43%
	Colores5	71,72%	89,90%	18,18%	42,11%	68,42%	26,31%
	Colores6	0,00%	100,00%	100,00%	0,00%	100,00%	100,00%
	Puntos1	15,15%	77,27%	62,12%	5,56%	50,00%	44,44%
	Puntos2	36,04%	88,29%	52,25%	3,85%	80,77%	76,92%
	Líneas1	71,21%	100,00%	28,79%	44,44%	100,00%	55,56%
	Líneas2	72,73%	88,81%	16,08%	46,15%	65,38%	19,23%
	FormaAleatoria1	98,03%	98,03%	0,00%	93,10%	93,10%	0,00%
	FormaAleatoria2	16,35%	25,96%	9,61%	4,17%	33,33%	29,16%
	Combinación1	67,72%	73,18%	5,46%	48,03%	44,09%	-3,94%
	Combinación2	90,51%	97,38%	6,87%	82,46%	87,72%	5,26%
	Combinación3	75,66%	84,65%	8,99%	45,00%	51,25%	6,25%
	Combinación4	0,00%	86,14%	86,14%	0,00%	62,79%	62,79%
Formato	Deformación1	95,58%	95,58%	0,00%	79,52%	79,52%	0,00%
	Deformación2	84,77%	89,95%	5,18%	62,65%	72,29%	9,64%
	Deformación3	46,58%	61,37%	14,79%	13,25%	36,14%	22,89%
	Deformación4	93,33%	93,33%	0,00%	66,67%	66,67%	0,00%
	Deformación5	100,00%	100,00%	0,00%	100,00%	100,00%	0,00%
	Deformación6	91,11%	100,00%	8,89%	77,78%	100,00%	22,22%
	Corte1	0,00%	100,00%	100,00%	0,00%	100,00%	100,00%
	Corte2	0,00%	95,00%	95,00%	0,00%	85,71%	85,71%

6. Conclusiones del proyecto

En este apartado del proyecto se presentarán todas aquellas conclusiones generales alcanzadas en el análisis que se ha llevado a cabo sobre las imágenes del estudio.

La siguiente figura muestra el porcentaje de mejora en el número de caracteres detectados correctamente por el OCR que hemos obtenido tras los procesados en cada una de las imágenes. Es decir, este porcentaje es la diferencia entre el porcentaje máximo alcanzado tras el estudio y el obtenido con la imagen original sin ningún procesamiento.

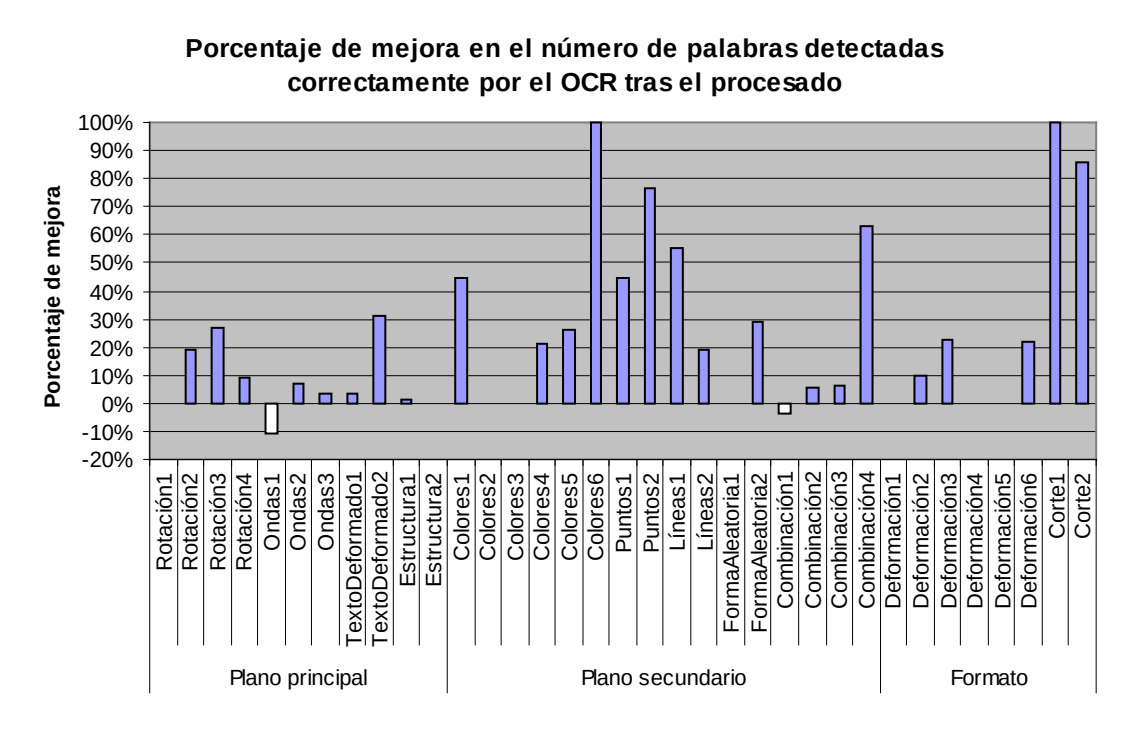


Entonces, con tan sólo observar el grafico previo, podemos llegar a la conclusión de que los mejores resultados de nuestro sistema se dan en aquellas imágenes que han sufrido modificaciones en el plano secundario, sin olvidarnos del caso de modificaciones por corte, dentro del tipo de cambios en el formato.

El hecho anterior se debe a que, en las alteraciones en el plano secundario o por corte, el texto presente en la imagen permanece sin distorsiones. Una vez que la alteración en el plano secundario o por corte ha sido eliminada, el texto es reconocible por el OCR sin mayores problemas.

Por lo tanto, siempre que se pueda nos va a interesar trabajar con este tipo de transformaciones ya que nos aseguran generalmente un mejor resultado de nuestro sistema de detección de SPAM.

La próxima imagen muestra igualmente la diferencia de porcentaje alcanzado tras el procesado y antes de éste, pero en este caso respecto al número de palabras detectadas correctamente.



Las conclusiones teniendo en cuenta el número de palabras son las mismas que hemos comentado para el número de caracteres, es decir, los mejores resultados se dan con las imágenes con alteraciones en el plano secundario y por corte, dentro de cambios en el formato.

Sin embargo, nos tenemos que detener en los casos de los archivos *Ondas1* y *Combinación1* ya que la tasa de palabras detectadas correctamente tras los procesados ha disminuido.

El hecho anterior se debe a que el número total de caracteres detectados de forma correcta puede aumentar, sin embargo, en cuanto uno de ellos ha fallado, la palabra está mal detectada. Es decir, se pueden detectar bien todos los caracteres presentes en una palabra exceptuando uno, lo que causará que la tasa de caracteres aumente, mientras que la de palabras permanecerá constante o disminuirá.

Por otra parte, existe otro factor que influye de gran manera en nuestro sistema, la calidad de la imagen. Hemos podido comprobar que el mismo tipo de perturbación es eliminada perfectamente en alguna de las imágenes, mientras que en otras no es posible ya que la calidad de la imagen hace sea imposible diferenciar el texto y de la perturbación.

Además otra de las conclusiones alcanzadas es que, a mayor tasa de acierto de la imagen original, es más complicado que aumente esta tasa tras los procesados ya que el procesado tiene menor rango de mejora sobre el que trabajar.

De cara a dar por finalizadas las conclusiones de proyecto, mencionar que siempre que podamos, intentaremos trabajar con imágenes con perturbaciones en el segundo plano ya que éstas aseguran mejores resultados en nuestro sistema de detección de SPAM gráfico con OCR.

7. Presupuesto

En este punto vamos a presentar el presupuesto necesario para llevar a cabo la realización de este proyecto.

El presupuesto estará dividido en dos partes: costes asociados al tiempo empleado en el desarrollo del proyecto y los costes de los materiales necesarios para éste.

Costes por tiempo:

Según el *Convenio Colectivo Nacional de empresas de ingeniería y oficinas de estudios técnicos* publicado en el BOE[1], el coste mínimo anual para Nivel 1(Licenciados y titulados) en el año 2009 asciende a 22937,51 €.

Por otro lado, asumimos las horas empleadas a lo largo de un mismo año son 1806 horas, de donde podemos despejar que el coste medio por hora de un ingeniero es de 12,70 €.

Coste salarial anual ingeniero:	22.937,51 €
Horas anuales:	1806
Coste por hora:	**12,70 €**

Una vez calculado el coste por hora, pasamos a estimar el tiempo empleado en el desarrollo del proyecto.

Duración del proyecto:	180 días
Tiempo medio empleado por día:	4 horas
Horas empleadas totales:	**720 horas**

Por lo tanto, el coste asociado al tiempo empleado es el siguiente:

Coste por hora:	12,70 €
Horas empleadas totales:	720 horas
Coste asociado al tiempo:	**9.144,00 €**

Costes por materiales:

La siguiente tabla muestra cada uno de los costes por materiales[2] que hemos tenido en este proyecto:

[1] http://www.boe.es/boe/dias/2009/03/28/pdfs/BOE-A-2009-5211.pdf#
[2] http://www.mathworks.es/store/default.do
http:// www.hp.com
http://www.telefonica.es
http://emea.microsoftstore.com/es/Microsoft/Office?WT.mc_id=OfficeOnline_ESES_Buy07_Ed

	Unidades	Precio unitario	Total
MATLAB & Simulink Student Version R2009a:	1	60,00 €	60,00 €
PC portátil HP Pavilion dv7-2150es:	1	899,00 €	899,00 €
Office Professional 2007:	1	649,00 €	649,00 €
SoftiFreeOCR:	1	0,00 €	0,00 €
Cuota mensual Conexión Internet Telefonica:	6	22,45 €	134,70 €
		Coste por materiales:	**1.742,70 €**

Coste final del proyecto:

En resumen, el presupuesto final para la ejecución de este proyecto es de 10.886,70 €, como podemos ver en el cuadro adjunto:

Coste asociado al tiempo:	9.144,00 €
Coste por materiales:	1.742,70 €
Coste total del proyecto:	**10.886,70 €**

8. Futuros proyectos a desarrollar

En el apartado anterior han quedado presentadas las conclusiones obtenidas con el desarrollo de nuestro proyecto Sin embargo, existen una serie de cuestiones que han quedado pendientes de tratar con anterioridad.

En nuestro sistema de detección de SPAM gráfico con OCR se pueden llevar a cabo varias automatizaciones en las distintas fases del proyecto.

Como hemos visto, las técnicas de procesado a aplicar sobre cada imagen dependen en gran manera de la perturbación que presente ésta. Pues bien, en nuestro proyecto, la clasificación de las imágenes de acuerdo a una de las categorías de modificaciones mencionadas en el apartado 3, se ha realizado visualmente. De esta forma, una mejora de nuestro sistema podría ser el introducir un nuevo bloque, el cual se encargue de detectar el tipo de cambios o modificaciones que ha sufrido cada imagen. Con la perturbación identificada, podríamos aplicar los procesados sin miedo a equivocarnos.

Recordar que las categorías de perturbaciones estudiadas son:

- Transformaciones en el plano principal
  - Rotación
  - Ondas
  - Texto deformado
  - Estructura

- Transformaciones en el plano secundario
  - Colores
  - Puntos
  - Líneas
  - Forma aleatoria
  - Combinación

- Transformaciones en el formato
  - Deformación
  - Corte

Por otra parte, durante el análisis práctico de las imágenes, han sido aplicados procesados que necesitaban una serie de parámetros de entrada para su funcionamiento. Estos parámetros de entrada eran introducidos por el usuario, de forma que el riesgo de un mal funcionamiento era mayor. Entonces, podríamos buscar el valor de estos parámetros automáticamente y reducir el porcentaje de errores, ya que el éxito únicamente dependería del procesado y en ningún caso del usuario.

Los parámetros que se podrían automatizar son los siguientes:

- Grados a girar la imagen en los procesados para eliminar la rotación.

- En el procesado para eliminar el ruido, el umbral que detecta cuando es ruido y cuando no lo es.

- En los filtrados por umbral, el valor que determina dicho umbral.

- Los clusters que queremos que formen parte de la imagen final en los procesados de imágenes a color.

- Tamaño de la máscara en filtrado de mediana, en los procesados para eliminar los puntos y para eliminar líneas.

- Los puntos de cortes en los procesados que evitan el corte en las imágenes.

Por lo tanto, todas estas automatizaciones presentadas anteriormente, tanto de parámetros de entrada como de categorías de perturbaciones, podrían convertirse en los puntos de partida de futuros proyectos a desarrollar, de manera que se mejore y quede más robusto nuestro sistema de detección de SPAM.

9. Bibliografía y enlaces WEB

Bibliografía

- Apuntes de la asignatura Aplicaciones del Tratamiento de Señales.
- Visión por Computador. Arturo de la Escalera. Edit. Prentice Hall
- Visión por Computador. Gonzalo Pajares y Jesús M. de la Cruz. Edit. Ra-Ma
- Ejercicios Resueltos de Visión por Computador. Gonzalo Pajares y Jesús M. de la Cruz. Edit. Ra-Ma
- Digital Image Processing. Rafael C. González y Richard E. Woods Edit. Pearson Internacional
- Help del programa Matlab

Enlaces WEB

- http://www.ironport.com/ar/technology/ironport_image_spam.html
- http://sunbeltblog.blogspot.com/2006/11/creative-image-spam.html
- http://www.borderware.com/pdfs/BW_ImageSpam_101106.pdf
- http://www.ceas.cc/2007/papers/paper-35.pdf
- http://www.ceas.cc/2007/papers/paper-40.pdf
- http://www.gfisoftware.de/es/whitepapers/why-bayesian-filtering.pdf
- http://www.thesmokesellers.com/?p=895
- http://decsai.ugr.es/gte/Seminars2003/JhgSpam.ppt.
- http://documentacion.irontec.com/tecnicas_anti_spam_EnpresaDigitala.pdf
- http://www.portalmundos.com/mundoinformatica/internet/antispam.htm
- http://alojamientos.us.es/gtocoma/pid/pid10/OCR.htm
- http://www.eset.es/download/files/docs/spam_hoy_ahora_y_siempre.pdf
- http://campusvirtual.uma.es/tdi/www_netscape/EstructuraContenidosI.php
- http://www.tsc.uc3m.es/imagine/Curso_ProcesadoBasico
- http://fcqi.tij.uabc.mx/docentes/esqueda/cursoimagenes.PDF
- http://www.scribd.com/doc/23371/Procesamiento-de-imagenes-con-Matlab
- http://www.mathworks.com/products/demos/image/color_seg_k/ipexhistology.html
- http://www.jstage.jst.go.jp/article/dsj/8/0/88/_pdf
- http://www.mathworks.com

10. Agradecimientos

Antes de dar por finalizado mi proyecto final de carrera quería dedicar unas cuantas líneas de éste a todos aquellos que han estado presentes durante su desarrollo.

A mis padres y a mi hermana que han estado siempre desde el principio de la carrera a mi lado. Tanto en los buenos como en los malos momentos están junto a mí. Gracias.

A mi novia Beatriz por apoyarme en todo el tiempo y comprenderme en cada instante. Además se ha encargado de darme fuerzas y ánimos en aquellos momentos en los que hacía falta. Gracias.

Al profesor D. Ángel Navia Vázquez por prestarme su ayuda en el desarrollo del proyecto y solventar todas esas dudas surgidas. Gracias.

Por último, a mis compañeros de trabajo en Ericsson que han hecho que este tiempo sea más agradable y llevadero. Gracias.

Apéndice: Materiales complementarios

En este apartado del proyecto vamos a detallar los programas desarrollados en Matlab para los distintos procesados llevados a cabo con las imágenes. Todos estos programas serán entregados en el CD que se adjunta junto a esta memoria.

En el CD anexo estarán también todas y cada una de las imágenes que han formado parte del proyecto, tanto los archivos originales como los archivos tras los distintos procesados. Además también será incluido el software utilizado para nuestra simulación del OCR, *SoftiFreeOCR*.

La organización de carpetas del CD es la siguiente:

Programas desarrollados en Matlab

Antes de comenzar con la explicación de cada uno de los programas que han sido creados para este proyecto, pasamos a ver cómo sería la ejecución de uno de estos programas en Matlab.

Para ejecutar uno de estos programas, en la pantalla principal de Matlab, tecleamos el nombre del programa junto con los parámetros de entrada que sean necesarios. Por ejemplo: *convertirByN('imagen.bmp')*, siendo *convertirByN*, el nombre del programa e *imagen.bmp*, el parámetro de entrada.

A continuación vamos a explicar brevemente los procesados que han sido creados para nuestro sistema de detección de SPAM.

- ***ajusteContraste('imagen')***
    - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.

- Parámetro de salida: imagen procesada que es guardada con nombre *imagen_cont*, en la misma carpeta en la que esté localizado el programa.
- Descripción: Procesado que se encarga de realizar un ajuste de contraste en la imagen original.

- ***cambiarDimensiones ('imagenDimensionesOrig','imagenDimensionesDest')***
  - Parámetros de entrada *imagenDimensionesOrig* e *imagenDimensionesDest*: imagen con las dimensiones que buscamos e imagen con las dimensiones a cambiar.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_dim*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de cambiar el tamaño de la imagen *imagenDimensionesDest* al tamaño de la imagen *imagenDimensionesOrig*.

- ***convertirByN ('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_ByN*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de transformar la imagen original a color a una nueva imagen en escala de grises.

- ***eliminarCorte ('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_corte*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de eliminar los cortes en el texto de la imagen. Es necesario que el usuario seleccione con el cursor los puntos superiores e inferiores del corte.

- ***eliminarLineas ('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_lineas*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de eliminar las líneas del plano secundario de la imagen. Es necesario que el usuario introduzca el tamaño de la máscara y el número de píxeles que hará de umbral para considerar que un punto pertenece a una línea a eliminar.

- ***eliminarOndas ('imagen')***

  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_ondas*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de eliminar las ondulaciones del texto presente en la imagen y alinear éste. Únicamente el usuario tiene que elegir si desea que el programa calcule automáticamente el nivel para el cambio de la imagen a color a una imagen binaria o bien es introducido por él.

- ***eliminarPuntos ('imagen')***

  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_puntos*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de eliminar los puntos del plano secundario de la imagen. Únicamente el usuario tiene que seleccionar el píxel de la imagen que se considerará umbral para la eliminación de los puntos.

- ***eliminarRotacion ('imagen')***

  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_rot*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de devolver la horizontalidad al texto, eliminando el giro que hayan sufrido las imágenes. El usuario se encarga de introducir los grados y el sentido que será girada la imagen.

- ***eliminarRotacionyBordesNegros ('imagen')***

  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_rot*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de devolver la horizontalidad al texto, eliminando el giro que hayan sufrido las imágenes y los bordes negros que el procesado anterior no eliminaba. El usuario se encarga de introducir los grados y el sentido que será girada la imagen.

- ***eliminarRuido('imagen')***

  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.

- Parámetro de salida: imagen procesada que es guardada con nombre *imagen_ruido*, en la misma carpeta en la que esté localizado el programa.
- Descripción: Procesado que se encarga de eliminar el ruido presente en la imagen. Es necesario que el usuario seleccione con el cursor el punto que será considerado como umbral en la detección de ruido.

- ***eliminarSubrayado('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_subray*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de eliminar el subrayado del texto que exista en la imagen. Es necesario que el usuario seleccione con el cursor el punto que será considerado como umbral en la eliminación del subrayado.

- ***filtradoMediana ('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_med*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que realiza un filtrado de mediana sobre la imagen. Únicamente el usuario tiene que introducir el tamaño de la máscara que se utilizará en el filtrado.

- ***igualacionHistograma('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_hist*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se genera una imagen procesada con el histograma igualado.

- ***procesadoImagenColor ('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_ _conjunto_clusters*, en la misma carpeta en la que esté localizado el programa. Además también son guardadas cada una de los clusters generados de la imagen.
  - Descripción: Procesado que se encarga de procesar las imágenes a color y generar clusters clasificados según su color. Únicamente el usuario tiene que elegir si desea que el programa calcule automáticamente el número de clusters

o bien sea introducido por él. La imagen final se compondrá de los clusters más interesantes seleccionados por el usuario.

- ***transformadaColor('imagen')***
  - Parámetro de entrada *imagen*: imagen RGB a procesar junto con su extensión.
  - Parámetro de salida: imagen YC_bC_r.
  - Descripción: Procesado que realiza un cambio de bases de RGB a YC_bC_r.

- ***umbral ('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_umb*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de aplicar un filtrado con umbral sobre la imagen. Únicamente el usuario tiene que seleccionar el píxel de la imagen que se considerará umbral.

- ***umbral PorSectores('imagen')***
  - Parámetro de entrada *imagen*: imagen a procesar junto con su extensión.
  - Parámetro de salida: imagen procesada que es guardada con nombre *imagen_sec*, en la misma carpeta en la que esté localizado el programa.
  - Descripción: Procesado que se encarga de aplicar filtrado con umbral sobre un área determinada de la imagen. El usuario tiene que seleccionar el píxel de la imagen que se considerará umbral, así como el área de la imagen sobre la que se aplicará éste.